New Rounding and Propagation Heuristics for Mixed Integer Programming

Bachelorarbeit bei
Prof. Dr. Dr. h. c. mult. M. Grötschel

vorgelegt von
Gregor Hendel[1]
Technische Universität Berlin
Fachbereich Mathematik

Berlin, 11. Mai 2011

[1]Konrad Zuse Institut für Informationstechnik Berlin, hendel@zib.de

Hiermit versichere ich die selbstständige und eigenhändige Anfertigung dieser Arbeit an Eides statt.

(Ort, Datum)	(Unterschrift)

Contents

Zusammenfassung

Die vorliegende Arbeit befasst sich mit Primalheuristiken für gemischt-ganzzahlige, lineare Optimierungsprobleme (*engl.: mixed integer program* (MIP)). Zahlreiche Optimierungsprobleme aus der Praxis lassen sich als MIP modellieren, Beispiele hierfür sind u. a. Optimierungsprobleme im öffentlichen Nah- und Fernverkehr, bei logistischen Fragestellungen oder im Bereich der Chip-Verifikation.

Das Lösen von MIP ist $\mathcal{NP}$-schwer und wird heutzutage meistens mit Hilfe von branch-and-bound-basierenden Algorithmen versucht. Das branch-and-bound-Verfahren profitiert unter Umständen von bereits frühzeitig zur Verfügung stehenden Lösungen, daher sind wir sehr an heuristischen Verfahren interessiert, die in der Praxis schnell eine gute Lösung für eine große Zahl an MIPs liefern und somit die Lösezeit des branch-and-bound-Verfahrens erheblich beschleunigen können.

Primalheuristiken sind Suchverfahren zum Auffinden zulässiger Lösungen eines MIP. Verschiedene Typen von Primalheuristiken sollen dabei den jeweiligen Bedarf des Anwenders zu unterschiedlichen Zeiten während der branch-and-bound-Suche decken. Während Start- und Rundeheuristiken zu Beginn des Löseprozesses eine große Rolle bei der Suche nach der ersten zulässigen Lösung haben, arbeiten Verbesserungsheuristiken auf schon bekannten Lösungen, um neue, bessere Lösungen zu produzieren.

Diese Arbeit beschäftigt sich mit Primalheuristiken, welche Teil des MIP-Lösers SCIP sind. Im ersten Kapitel werden nach der Erarbeitung grundlegender Definitionen viele der durch Tobias Achterberg und Timo Berthold in SCIP integrierten heuristischen Verfahren vorgestellt und kategorisiert. Auf dieser Grundlage bauen dann die Kapitel 2–4 der Arbeit auf. In diesen werden drei zusätzliche Heuristiken vorgestellt, im Einzelnen sind dies ZI round, eine Rundeheuristik, welche zuerst in [Wal09] beschrieben wurde, außerdem eine 2-opt-Heuristik für MIP und eine neue Startheuristik, *Shift and Propagate*.

Großer Wert wird in jedem Kapitel auf die algorithmische Beschreibung der Heuristiken gelegt, die stets anhand von motivierenden Beispielen eingeführt und anhand von Pseudocode-Algorithmen begleitet werden. Zusätzlich enthält jedes Kapitel Auswertungen der mit den neuen Heuristiken gemessenen Ergebnisse von SCIP. Eine kurze Zusammenfassung in Kapitel 5 schließt diese Arbeit ab.

Die Implementation der drei vorgestellten Heuristiken wurde vom Autor dieser Arbeit vorgenommen.

Danksagungen

Ich möchte mich bei meiner Mutter bedanken, die sich stets mit aller Kraft dafür einsetzt, dass ihren Kindern mehr Wege offen stehen als ihr selbst.

Ein ganz besonderer Dank geht an Timo Berthold für die Einarbeitung, die umfangreichen Tipps und die Hilfestellungen zu jeder Zeit während meiner Arbeit hier am ZIB.

Ich möchte mich bei Richard Sieg, Matthias Miltenberger und Jonatan Krolikowski für ihr korrigierendes Auge beim Lesen einer ersten Fassung dieser Arbeit bedanken. Ein LaTeX-Dank soll an dieser Stelle an Jonas Schweiger gerichtet sein, dessen Wissen zu diesem Thema ich gerne und stets in Anspruch nehme.

Ganz herzlich möchte ich mich auch bei Herrn Prof. Dr. Grötschel für die Betreuung der vorliegenden Arbeit bedanken.

Zu guter Letzt geht ein besonderer Dank an meine Freundin Katharina, die mich stets hat motivieren können und deren wunderbare Liebe mir hoffentlich noch lange erhalten bleibt.

Introduction

Many practically relevant problems can be formulated in terms of a *mixed integer programming*(MIP) model. MIP denotes the optimization of a linear objective function under a certain number of linear side constraints including the need for some of the involved variables to take integral solution values. Applications of MIP based optimization can be found in the area of public transit(see, e. g., [BGJ10]), scheduling, automatic vehicle routing, network design, etc.

From a complexity point of view, MIP solving is known to be $\mathcal{NP}$-hard and most commonly tried to be solved via branch-and-bound based algorithms. branch-and-bound algorithms benefit from early and good feasible solutions of a MIP in various ways.

Primal heuristics are aimed at finding new solutions during the MIP solving process. There are different types of primal heuristics: while start heuristics are particularly valuable to find an early solution, improvement heuristics hopefully drive a given solution further towards optimality.

This thesis focusses on primal heuristics which are part of the MIP-solving framework SCIP [Ach07]. The first chapter comes with basic definitions and a brief description of SCIP and the test set which we used. The remainder of the first chapter is an overview of the existing heuristics in SCIP which have been implemented by Achterberg [Ach07] and Berthold [Ber06].

In the following chapters we introduce three new heuristics which apply rounding or propagation techniques for their specific purpose, namely the new rounding heuristic ZI round, taken from [Wal09], a 2-opt improvement heuristic for MIP and the propagation heuristic Shift and Propagate. It is characteristic of all three heuristics that they mainly apply computationally inexpensive algorithms.

Each of them is presented in an own chapter, starting with an algorithmic description, followed by implementational details. All chapters close with a discussion of the computational results obtained with the respective implementations in SCIP.

Chapter 1

MIP problems and Primal Heuristics

1.1 Definitions

This section gives precise definitions of the necessary terms used throughout this thesis. We start with the definition of a *mixed integer program* (MIP).

Definition 1. *Let $m, n \in \mathbb{N}$, $\tilde{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$, $\mathcal{I}, \mathcal{C} \subseteq \mathcal{N}$ be a partition of the variable index set $\mathcal{N} := \{1, \dots, n\}$, $l, h \in \tilde{\mathbb{R}}^n$ be bound vectors, $A \in \mathbb{R}^{m,n}$ a real matrix, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. Now we can formulate a* mixed integer program(MIP) *P as follows:*

$$
\begin{aligned}
\text{Minimize} \quad & c^T x && && (1.1)\\
s.\,t. \quad & Ax \le b && && (1.2)\\
& l \le x \le h && && (1.3)\\
& x_j \in \mathbb{Z} && \textit{for } j \in \mathcal{I} && (1.4)\\
& x_j \in \mathbb{R} && \textit{for } j \in \mathcal{C} && (1.5)
\end{aligned}
$$

For all variables $j \in \mathcal{N}$, l_j and h_j denote its lower and upper bound, respectively. A variable $j \in \mathcal{I}$ will be referred to as *integer* variable. Integer variables are subdivided in the class of *binary* variables $j \in \mathcal{B} \subseteq \mathcal{I}$ with $l_j = 0$ and $h_j = 1$ and in *general integer* variables $j \in \mathcal{I} \setminus \mathcal{B}$. In case we have to distinguish between finite or infinite variable bounds, we denote by

$$D_j := \{d \in \mathbb{Z} | \, l_j \le d \le h_j\} \tag{1.6}$$

the *domain* of variable j in ordert to simplify the notation.

MIPs can be classified w. r. t. the occurence of continuous, general integer and binary variables;

$\mathcal{C} = \mathcal{N}$ only continuous variables form a *Linear Program* (LP)

$\mathcal{I} = \mathcal{N}$ an *Integer Program* (IP)

$\mathcal{I} \ne \emptyset \ne \mathcal{C}$ a *Mixed Integer Program* (MIP)

$\mathcal{B} = \mathcal{N}$ a *Binary Program* (BP)

We will call c the *objective function* or just objective of P. The m inequalities of the form (1.2) will be called (LP-)*rows* of P. Every column A_j of A corresponds to a variable $j \in \mathcal{I} \cup \mathcal{C}$. For all matrix calculations we will use data structures which only take nonzero entries of A into account. We will use $|A^{\neq 0}|$ and $|A_j|$ to denote the number of nonzero entries of A and A_j, respectively.

For a problem P with integer variables $\mathcal{I} \neq \emptyset$, the *LP-relaxation* P^*

$$\text{Minimize} \quad c^T x \text{ s.t. } x \in \underbrace{\{y \in \mathbb{R}^n \,|\, Ay \leq b,\, l \leq y \leq h\}}_{\sigma_P} \tag{1.7}$$

is obtained by disregarding the integrality conditions 1.4. Therefore, P^* is an LP. The set σ_P of feasible points for P^* is called the *polyhedron* of P. All solutions feasible for P are also feasible for P^*, i. e.,

$$D_P \subseteq D_{P^*}. \tag{1.8}$$

We will call a solution $z \in \sigma_P$ *LP-feasible* and *IP-feasible* if it is LP-feasible and satisfies the integrality conditions 1.4. z is *optimal* for a MIP P if it is IP-feasible and its objective value satisfies

$$c^T z = \min\{c^T x \,|\, x \text{ IP-feasible for } P\}. \tag{1.9}$$

The objective value $c^T z^*$ of an optimal solution z^* of P^* is a valid *lower bound* to the objective value $c^T z_{\text{opt}}$ of an optimal IP-feasible solution z_{opt} of P because of (1.8). The objective value $c^T z$ of every IP-feasible solution z is a valid upper bound to the objective value of an optimal solution. Whenever we consider the relaxation solution z^* we implicitly assume that z^* is optimal for P^*.

Definition 2. *Let z^* be the solution of the LP relaxation of a given MIP problem P. We define*

$$F := \{j \in \mathcal{I} : z_j^* \notin \mathbb{Z}\} \tag{1.10}$$

to be the set of integer variables which take fractional solution values in z^*. For a variable $j \in \mathcal{I}$ we define the fractionality *of j by*

$$f_j^{\min} := \min\{f_j^+, f_j^-\} \tag{1.11}$$

for the partial fractionalities

$$f_j^+ := \lceil z_j^* \rceil - z_j^* \text{ and } f_j^- := z_j^* - \lfloor z_j^* \rfloor, \tag{1.12}$$

and the integer infeasibility *of the entire solution as the sum over all variable fractionalities,*

$$f^{\min}(z) = \sum_{j \in \mathcal{I}} f_j^{\min}. \tag{1.13}$$

Furthermore, the left hand term of the i-th LP row

$$a_i^T z \leq b_i \tag{1.14}$$

is the activity *of i w. r. t. z, and the difference of activity and right hand side*

$$s_i := b_i - a_i^T z \tag{1.15}$$

is called slack *of i.*

A solution z is LP-feasible if and only if $s_i \geq 0$ for all rows $1 \leq i \leq m$. For an LP-feasible solution z it holds

$$z \text{ IP-feasible } \Leftrightarrow F = \emptyset \Leftrightarrow f^{\min}(z) = 0. \tag{1.16}$$

LPs can be solved in polynomial running time [AS80]. In praxis, LPs are mostly solved by using the *Simplex* method (see, e. g., [ES00, ch. 1] or [Chv83, ch. 2]).

MIP solving, however, is $\mathcal{NP}$-hard [CLRS01]. MIPs are usually solved by *branch-and-bound* based algorithms [Ach07]. Given the (not IP-feasible) solution z^* of the LP relaxation P^* of a MIP P, one of the integer variables $j \in F$ with fractional solution value z_j^* is selected by a branching rule. Every IP-feasible solution z either satisfies $z_j \leq \lfloor z_j^* \rfloor$ or $z_j \geq \lceil z_j^* \rceil$. Hence two subproblems of P are created by adjusting the bounds $l_j \leq x_j \leq h_j$ of the variable either to

$$h_j \leftarrow \lfloor z_j^* \rfloor \tag{1.17}$$

or to

$$l_j \leftarrow \lceil z_j^* \rceil. \tag{1.18}$$

Two subproblems are created by this bounding step, corresponding to two new nodes (branches) in the branch-and-boundtree. P is consecutively subdivided into smaller subproblems.

We call a solution z *incumbent* if z is IP-feasible for a MIP P and has the minimum objective value $c^T z$ of all IP-feasible solutions found so far. The objective $c^T z$ of an incumbent z is the *primal bound*. For a given primal bound $c^T z$ and a lower bound $\underline{c}$ to the objective value $c^T z_{\text{opt}}$ of an optimal solution z_{opt} we call

$$\Delta := \begin{cases} \frac{|c^T z - \underline{c}|}{|\underline{c}|}, & |\underline{c}| > 0 \\ 0, & c^T z = \underline{c} = 0 \\ \infty, , & \text{else} \end{cases} \tag{1.19}$$

the *primal-dual gap*. $\Delta = 0$ causes the branch-and-bound search to terminate because an incumbent with a primal-dual gap of zero is proven to be optimal.

The early knowledge of a feasible primal solution confirms the feasibility of the model and delivers a primal bound and a primal-dual gap. We also obtain an upper bound to the objective value of an optimal solution. Upper bounds help to prune regions of the search tree and can thus reduce the number of required nodes to optimality. Furthermore, a feasible start solution is required by all improvement heuristics(see sec 1.4.4). If the solution has been made available while processing the root node, some of the integer variables of the model might be fixed due to their reduced costs (for further information about *reduced cost fixing*, see [Ach07]).

Hence a solution at an early stage of the solving process can be beneficial to reduce the gap to optimality early in the solving process.

The last definition in this section defines the *geometric mean*, which we want to use for benchmarks.

Definition 3. *Let $n \in \mathbb{N}$, and $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ be a set of real values. We call*

$$\varnothing_X := \sqrt[n]{\prod_{i=1}^{n} \max\{x_i, 1\}}$$

the geometric mean *of X.*

The modification to consider the maximum of each x_i and 1 avoids giving values near zero much relevancy. If, e. g., the absolute solving time on an instance was improved from 0.02 seconds to 0.01 seconds, due to Definition 3 we do not consider such an improvement equal to a time reduction from 10 hours to 5 hours on a different instance.

1.2 SCIP - Solving Constraint Integer Programs

The solver software used throughout this thesis is SCIP [Ach07]. The development of SCIP was initiated by Tobias Achterberg at the *Konrad-Zuse-Institut Berlin* (ZIB) in 2002. SCIP is designed to solve Constraint Integer Programs (CIPs) [Apt03]. SCIP can be used as a stand-alone MIP solver using a branch-and-bound strategy. The branch-and-bound search is supported by *separation* algorithms [Wol06], *propagation routines* and *primal heuristics*, which are delivered as independent plugins. Current developers of SCIP are Timo Berthold, Gerald Gamrath, Gregor Hendel, Stefan Heinz, Marc Pfetsch, Stefan Vigerske, Michael Winkler, and Kati Wolter. Written in the programming language C, SCIP is freely available under an academic licence at `http://scip.zib.de`.

1.3 Test set

The benchmark set used throughout this thesis consists of 163 MIP problems taken from three public MIP libraries: the Miplib 3.0 [BCMS98], the Miplib 2003 [AKM06], and the MIP collection of Mittelmann [Mit03]. In particular experiments we decided to exclude certain problems if they caused, e. g., numerical troubles or if they prematurely ran out of of memory so that no statistics were available. Exclusions of that kind are documented in the respective sections. We ran the tests with SCIP version 2.0.1 and ILOG CPLEX version 12.2 [cpl] as the underlying LP solver.

1.4 Primal heuristics in SCIP

Primal heuristics for MIP are algorithms which aim at finding an IP-feasible solution. SCIP already contains a set of primal heuristics. In this section, we will give an overview of the existing heuristics. They are subdivided into four categories, *rounding*, *diving*, *objective diving* and *improvement* heuristics in accordance with the categorization in [Ach07]. Rounding heuristics, e. g., are designed to drive a given LP-feasible solution towards IP-feasibility whereas improvement heuristics are supposed to improve the solution value $c^T z$ of an IP-feasible incumbent z. Therefore, rounding heuristics are particularly valuable to find start solutions early during search. All primal heuristics presented here can be found in [Ach07] or [Ber06].

Heuristics can have a large effect on the branch-and-bound solving process. Figure 1.1 shows the benchmark results of SCIP with disabled heuristics and SCIP with default settings. For this comparison we consider a total number of 118 instances from the test set for which all solvers terminated with an optimal solution within the time limit of 30 min. and the memory limit of 2 GB.

We see that applying the SCIP heuristics reduces the geometric mean number of solving nodes $\varnothing_{\text{nodes}}$ by a factor of 2.5 from 890.59 to 354.03. The geom. mean solving

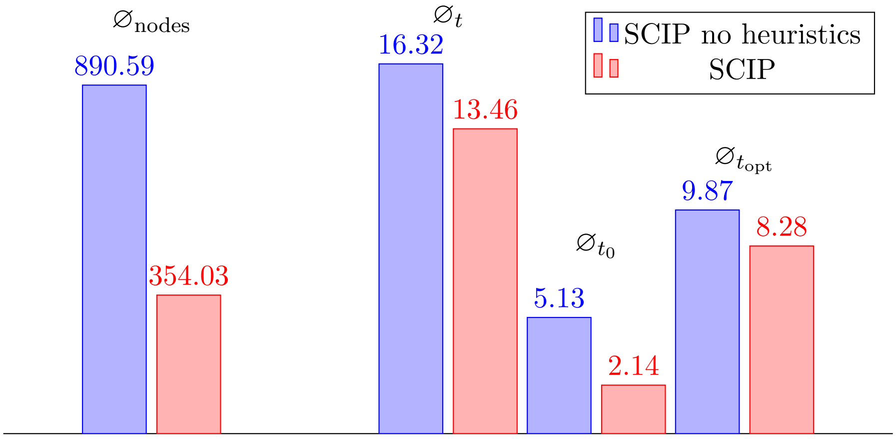


Figure 1.1: The impact of heuristics on the number of solving nodes and the solving time of SCIP 2.0.1. The bars show the geometric mean of the number of solving nodes $\varnothing_{\text{nodes}}$ and geom. means concerning the overall solving time t, the time t_0 until the first solution was found, and t_{opt} the time until the optimal solution was found.

time $\varnothing_t$ was reduced by 17.5 % from 16.32 sec to 13.46 sec The time until the first feasible solution is found is decreased by 59 % from 5.13 sec to 2.14 sec, and the time until the optimal solution was reduced by 17.2 % from 9.87 sec to 8.28 sec

Trivial heuristic

Trivial is a straightforward start heuristic which generates and checks four different solutions for IP-feasibility for P.

- the *zero solution* $z_j \leftarrow 0$ for all $j \in \mathcal{N}$,
- the *upper bound solution* $z_j \leftarrow h_j$ for all $j \in \mathcal{N}$,
- the *lower bound solution* $z_j \leftarrow l_j$ for all $j \in \mathcal{N}$,
- the *lock solution* $z_j \leftarrow h_j$ for all $j \in \mathcal{N}$.

There are many combinatorial optimization problems for which at least one of those "trivial" solutions is feasible, e. g., set covering and set partitioning instances for which the upper bound solution and the zero solution are feasible, respectively. The trivial heuristic can be already applied before the presolving stage of SCIP.

1.4.1 Rounding heuristics

Rounding the fractional solution values of an LP-feasible solution can be a quick way to obtain an IP-feasible solution.

The heuristics in this section have an LP-feasible solution z^* with $F \neq \emptyset$ as input and try to apply specific rules to determine a rounding for all variables $x_j \in F$. We present the *simple rounding*, the *rounding* and the RENS heuristic in this section. Furthermore we give an overview of *shifting* and *integer shifting* although they are no "pure" rounding heuristics, because they apply shifts of already integral solution values. The new ZI Round heuristic will be subject to the chapter 2. The *octane* heuristic is not activated by default in SCIP version 2.0.1. Therefore, it is not discussed in this thesis. We refer to Balas et al. [BCD⁺01] for the algorithmic background or to Berthold [Ber06] for further information about the SCIP implementation of octane.

Simple rounding

The *simple rounding* heuristic uses the notion of *up* and *down locks* to determine if a rounding is possible. In our standard notation, rounding up a fractional solution value z_j^*, $j \in F$ might violate the respective row $a_i^T x \leq b$ if $a_{ij} > 0$. In this case, a_{ij} is considered an up lock of x_j. The number of up locks of a variable x_j is counted by $\overline{\Lambda}_j$. Similarly, the number of down locks $\underline{\Lambda}_j$ is the number of rows for which A_j has a negative coefficient $a_{ij} < 0$. A variable is said to be *trivially roundable* if $\underline{\Lambda}_j = 0$ or $\overline{\Lambda}_j = 0$. Simple rounding processes the variables with fractional solution values and applies a trivial rounding if possible (see Algorithm 1).

Algorithm 1: Simple round algorithm

Input : a MIP instance P, an LP-feasible solution z of P with $F \neq \emptyset$.
Output : an IP-feasible solution $\hat{z}$ of P or NULL

$\hat{z} \leftarrow z$;
foreach $j \in F$ **do**
  if $\underline{\Lambda}_j = 0$ **then**
    $\hat{z}_j \leftarrow \lfloor z_j \rfloor$;
  else if $\overline{\Lambda}_j = 0$ **then**
    $\hat{z}_j \leftarrow \lceil z_j \rceil$;
  else
    Stop;
    return NULL;
end
return $\hat{z}$;

Rounding

Like simple rounding, the *rounding* heuristic takes variable locks $\underline{\Lambda}_j, \overline{\Lambda}_j$ of a variable with fractional relaxation value z_j^* into account. If no LP row is violated, the algorithm iterates over the fractional variables $j \in F$ and applies a rounding in the direction of fewer locks.

$$\hat{z}_j \leftarrow \begin{cases} \lfloor z_j \rfloor, & \text{if } \underline{\Lambda}_j \leq \overline{\Lambda}_j, \\ \lceil z_j \rceil, & \text{else} \end{cases} \tag{1.20}$$

The rationale of the procedure is to avoid roundings which will violate LP rows more likely. If, however, the current rounding solution $\hat{z}_j$ violates LP rows, we select one of the violated rows and try to decrease the violation. This is performed by selecting a variable from the set F whose rounding can drive the row activity towards feasibility but has the least number of locks in that direction amongs all variables of that kind.

Shifting

The *shifting* heuristic follows the same rounding rule like the rounding heuristic. In case of a row violation for which no feasibility recovering rounding can be found, a variable $j \in \mathcal{I} \setminus F$ with integral solution value z_j is tried to be shifted for the repairing step.

Integer shifting

Integer shifting relaxes continuous variables from P and obtains a transformed problem $\hat{P}$. It then applies the shifting algorithm to $\hat{P}$. If the obtained solution $\hat{z}$ is potentially feasible, the heuristic fixes all integer variables

$$x_j \leftarrow \hat{z}_j,\ j \in \mathcal{I}, \tag{1.21}$$

and solves the resulting LP. If a feasible solution is found, this is also a solution to P. Further information about relaxing continuous variables is given in 4.2.1 on page 42.

RENS

The *relaxation enforced neighborhood search* (RENS) heuristic solves a sub-problem of P by adjusting the bounds of variables with fractional solution values z_j^* in the solution of the LP-relaxation P^*. Given a solution z^* of P^* with $F \neq \emptyset$, RENS solves a sub-MIP P_{RENS} which is defined as

$$\begin{aligned}
P_{\text{RENS}}: \quad \text{Minimize} \quad & c^T x && && (1.22)\\
\text{s.t.} \quad & Ax \leq b && && (1.23)\\
\lfloor z_j^* \rfloor \leq x_j & \leq \lceil z_j^* \rceil && \text{for } j \in \mathcal{I} && (1.24)\\
x_j & \in \mathbb{Z} && \text{for } j \in \mathcal{I} && (1.25)\\
x_j & \in \mathbb{R} && \text{for } j \in \mathcal{C}. && (1.26)
\end{aligned}$$

Note that the condition (1.24) fixes all variables with integral solution values z_j^*. P_{RENS} is only solved if it is potentially easier than the original problem P, i. e., if at least an appropriate ratio of variables is fixed. The feasible area of P_{RENS} contains all solutions which are obtainable from z^* via a pure rounding strategy.

1.4.2 Diving heuristics

Like rounding heuristics, diving heuristics take the solution z^* of the LP relaxation as input to (hopefully) generate IP-feasible solutions. Instead of a direct rounding, the MIP problem is modified by changing the bounds of a variable $l_j \leq x_j \leq h_j$, $j \in F$ with fractional solution value z_j^* to either $\hat{h}_j \leftarrow \lfloor z_j^* \rfloor$ or $\hat{l}_j \leftarrow \lceil z_j^* \rceil$. The bound change causes z^* to be infeasible for the LP relaxation of the obtained MIP $\hat{P}$. As the (optimal) dual basis of z^* for P^* stays dual feasible for $\hat{P}^*$, the obtained LP relaxation can be efficiently resolved by using the *Dual Simplex Algorithm* [Sch86]. The consecutive bounding and resolving of the LP relaxation is called diving. Diving either stops with a feasible solution for P, or the explored diving tree is guided towards infeasibility. In the latter case, a 1-level backtracking can be applied. The algorithm chooses the remaining bounding direction for the variable and continues. Diving heuristics also apply simple rounding steps and the execution of domain propagation routines to reduce the number of necessary LP iterations by fixing variables or reducing domains.

Diving heuristics vary in their respective selection of a variable and a bounding direction. Guided Diving was originally proposed as a child node selection rule for the branch-and-bound search in [DRL04]. Line Search Diving is adapted from another child node selection rule [Mar99]. In general, every branching strategy for the branch-and-bound search can be used inside a diving heuristic, as well. The bounding step of a variable at each diving depth is also called a *hard rounding*.

Coefficient diving

Coefficient diving compares all fractional valued variables of the solution z^* w. r. t. to their smaller number of up and down locks

$$\Lambda_j^{\min} := \min\{\overline{\Lambda}_j, \underline{\Lambda}_j\},\ j \in F. \tag{1.27}$$

The variable which minimizes $\min\limits_{j \in F} \Lambda_j^{\min}$ is selected and bounded into the direction which has the fewer number of locks.

Fractionality diving

The variable is selected which has the minimum fractionality (see Definition 2)

$$\min_{j \in F} f_j^{\min} \tag{1.28}$$

and bounded into the direction of its closer integer neighbor.

Guided diving

Guided diving requires an incumbent solution z. The variable selection rule is the same as in Fractionality Diving. The selected variable is then bounded into the direction of its solution value z_j in the incumbent solution, i. e., upwards, if $z_j^* < z_j$, else downwards.

Line search diving

Line search diving compares the solution (z_{R}^*) of the root node LP relaxation P_0^* and the LP relaxation solution at the current node z^*. Those variables $j \in F$ are considered for which $z_j^* \neq (z_{\mathrm{R}}^*)_j$. Depending on the sign of $z_j^* - (z_{\mathrm{R}}^*)_j$, every variable has an assigned *distance ratio*

$$Q_j := \begin{cases} \frac{z_j - \lfloor z_j \rfloor}{(z_{\mathrm{R}}^*)_j - z_j^*}, & \text{if } z_j^* < (z_{\mathrm{R}}^*)_j \\ \frac{\lceil z_j \rceil - z_j}{z_j^* - (z_{\mathrm{R}}^*)_j}, & \text{if } z_j^* > (z_{\mathrm{R}}^*)_j. \end{cases} \tag{1.29}$$

Q_j rates the distance of the solution value z_j to the next integer in the direction of the line from $(z_{\mathrm{R}}^*)_j$ to z_j compared to the distance $|(z_{\mathrm{R}}^*)_j$ - $z_j|$. The variable with minimum distance ratio Q_j is bounded in the corresponding direction.

Pseudocost diving

Pseudocost diving combines the fractionality of a variable and its *pseudocosts* values [Ach07, 5.3] in a single measure to select a bounding direction. Pseudocosts Π_j^+, Π_j^- measure the expected objective value change per unit if variable x_j is bounded upwards and downwards, respectively. The bounding direction of a variable is chosen by the following rule:

1. if $|z_j - (z_{\mathrm{R}}^*)_j| \geq 0.4$ the bounding direction is downwards if $z_j < (z_{\mathrm{R}}^*)_j$, else upwards.
2. if 1 is not fulfilled, the fractionality of the variable is considered. If $f_j^{\min} < 0.3$, the variable is bounded in the direction of that minimum.
3. in case that 1 and 2 do not lead to a decision, the bounding is chosen w. r. t. the smaller pseudocosts, i. e., downwards if $\Pi_j^+ > \Pi_j^-$, else upwards.

Once the direction is chosen, the variable is bounded which maximizes

$$\sqrt{f_j^+} \cdot \frac{1+\Pi_j^+}{1+\Pi_j^-} \text{ (downwards) or } \sqrt{f_j^-} \cdot \frac{1+\Pi_j^-}{1+\Pi_j^+} \text{ (upwards).} \tag{1.30}$$

Vector length diving

Vector length diving chooses the bounding direction w. r. t. the sign of the objective coefficient c_j. Variables with nonnegative coefficient $c_j > 0$ are bounded upwards and those with negative coefficient are bounded downwards. Note that the bounding step will always deteriorate the objective value $c^T z^*$ of the current relaxation solution. With the notation of partial fractionalities f_j^+, f_j^-, the variable is chosen with the smallest ratio

$$\frac{f_j^+ c_j}{|A_j|+1} \text{ upwards or } -\frac{f_j^- c_j}{|A_j|+1}. \tag{1.31}$$

Ideally, a variable with a small deteriorating impact on the objective value is chosen which covers a large number of LP rows.

1.4.3 Objective diving heuristics

In constrast to diving heuristics, *objective diving heuristics* modify the objective coefficients of a selected variable instead of their bounds. The modification is supposed to drive the variable towards its lower or upper bound. Omitting bound changes has the advantage that the solution obtained by the previous iteration remains feasible for the modified problem. The modifications of the objective functions are also called *soft roundings*. Domain propagation routines, however, cannot be applied because the domains of the variables stay unchanged. We present the *objective pseudocost heuristic*, the *root solution diving* and the *feasibility pump* heuristic. The latter was originally proposed in [FGL05], generalized to MIP in [BFL05] and later modified by [AB07].

Objective pseudocost diving

Objective pseudocost diving applies the same rules for variable and direction selection as *pseudocost diving*. The objective coefficient c_j of the selected variable $j \in F$ is then modified to

$$\hat{c}_j \leftarrow \begin{cases} 1000 \cdot (d+1) \cdot |c_j|, & \text{(downwards)} \\ -1000 \cdot (d+1) \cdot |c_j|, & \text{(upwards)} \end{cases} \tag{1.32}$$

to force x_j to the desired bound after the LP resolve. d denotes here the current depth in the diving tree. The heuristic keeps track of all intermediate modifications to the original problem P to avoid cycling. If a variable is selected a second time it is bounded into the opposite direction of the previous modification instead. If the variable could not be driven to its favorable, e. g., lower bound by the objective coefficient modification, it is likely that the lower bound is infeasible for the problem. Thus, the heuristic applies a hard rounding upwards in that case. If the hard rounding leads to an infeasibility, the heuristic terminates.

Root solution diving

Root solution diving selects variables and directions like the *Line Search Diving* heuristic. At each new diving node, all objective coefficients are scaled by 0.9. Furthermore, the objective coefficient c_j of a selected variable $j \in F$ is either increased by $0.1 \cdot \max\{c^T z^*, 1\}$ where $c^T z^*$ is the objective value of the LP relaxation solution z^* from which the dive was started. In order to avoid cycling the heuristic remembers the number of soft up and down roundings s_j^+ and s_j^-. If the two numbers differ by more than 10, a hard rounding step is applied in the direction of the larger number of soft roundings. Variables whose solution value becomes integral during the dive are fixed to this value.

Feasibility pump

The *feasibility pump* heuristic derives an artificial objective function as distance measure from the rounded vector $[z^*]$ of the LP-feasible relaxation solution z^*. The rounding $[\cdot]$ of an LP-feasible solution z is defined as

$$[z]_j := \begin{cases} \lfloor z_j + 0.5 \rfloor, & \text{for } x_j \in \mathcal{I} \\ z_j, & \text{else.} \end{cases} \tag{1.33}$$

Let $d(x, y)$ denote the distance of two vectors x, y w. r. t. the integers $\mathcal{I}$, i. e.,

$$d(x, y) := \sum_{j \in \mathcal{I}} |x_j - y_j|. \tag{1.34}$$

Note that IP-feasible solutions will not be changed by rounding. In case of z^*, its rounded vector $[z^*]$ satisfies the integrality conditions 1.4 of P but is not necessarily LP-feasible.

Feasibility pump consecutively solves a sequence of LPs with the objective to find an LP-feasible point $\tilde{x}^k \in \sigma_P$ with minimum distance

$$d(\tilde{x}^k, [\tilde{x}]^{k-1}) = \min_{x \in \sigma_P} d(x, [\tilde{x}]^{k-1}) \tag{1.35}$$

to the rounded solution vector $[\tilde{x}]^{k-1}$ from the previous iteration, with $\tilde{x}^0 := z^*$. The heuristic terminates if an IP-feasible point $\tilde{x}$ was found or an iteration limit is exceeded.

As proposed in [AB07], feasibility pump in SCIP combines the distance measure d and the original objective coefficients c of P to a convex combination

$$d_\alpha(x, [\tilde{x}]^{k-1}) := (1 - \alpha) \cdot d(x, [\tilde{x}]^{k-1}) + \alpha \cdot \xi \cdot c^T x, \tag{1.36}$$

with a factor $\alpha = \alpha_k = \phi \alpha_{k-1}$ which is decreased geometrically in each iteration by a fixed parameter $\phi < 1$, and a constant $\xi = \frac{|\mathcal{I}|}{\|c\|}$. With decreasing α, (1.36) emphasizes the search for feasibility and slowly fades out the objective criterion.

1.4.4 Improvement heuristics

Improvement heuristics try to modify the current incumbent in order to obtain IP-feasible solutions with a better objective value. Many of these methods methods rely on the solving of a sub-MIP after the fixation of a set of variables. Parameter decisions are required to omit the solve of sub-MIPs which are not substantially easier than the original MIP problem.

Crossover

Crossover in SCIP takes three IP-feasible solutions z^1, z^2, z^3 and solves a sub-MIP obtained by fixing all discrete variables for which the respective solution values agree, i. e., all $j \in \mathcal{I}$ for which $z_j^1 = z_j^2 = z_j^3$. Crossover processes the three currently best IP-feasible solutions or, if this was already performed, three randomly selected solutions.

Local branching

Local branching [FL03] investigates a neighborhood of an incumbent solution z of a MIP instance P with non-empty binary variable set $\mathcal{B} \neq \emptyset$. For a positive integer $k > 0$ the solutions in the neighborhood $\mathcal{N}(z, k)$ are feasible for P and satisfy the additional *local branching constraint*

$$\sum_{j \in \mathcal{B}} |x_j - z_j| \leq k. \tag{1.37}$$

$\mathcal{N}(z, k)$ contains all solutions of P whose binary solution values are different from z for at most k variables $j \in \mathcal{B}$. With the notation

$$S := \{j \in \mathcal{B} \mid z_j = 1\} \tag{1.38}$$

denoting the subset of binary variables with solution value 1, (1.37) can be easily linearized [FL03] to

$$\sum_{j \in S} (1 - x_j) + \sum_{j \in \mathcal{B} \setminus S} x_j \leq k \tag{1.39}$$

and added as LP row to define a sub-MIP P_{Local} with domain $\mathcal{N}(z, k)$. The choice of k determines the size of the sub problem and can thus influence the its solving time. Fischetti, Matteo and Lodi [FL03] suggest a $k \in [10, 20]$. In SCIP, k is adaptively decreased by 50 % if the resulting sub MIP could not be solved within a given node limit, and increased if the heuristic could not find a better solution than z.

Mutation

Mutation fixes a set of discrete variables of the current incumbent z to their values and solves the resulting sub-MIP. The SCIP implementation fixes 80 % of the discrete variables of P on a random basis.

One opt

The *one opt* heuristic is based on the fact that the incumbent z can be improved if a variable with objective coefficient $c_j > 0$ can be feasibly decreased or a variable with negative objective coefficient can be feasibly increased. The heuristic determines maximum shifting values δ_j in the preferable direction w. r. t. the objective as described above for all variables $j \in \mathcal{I}$. Those variables with a positive shifting value $\delta_j > 0$ are then shifted in nonincreasing order of their impact $|\delta_j \cdot c_j|$ on the objective.

RINS

Relaxation induced neighborhood search (RINS) solves a sub-MIP which results from the comparison of the current incumbent z and the (not IP-feasible) solution z^* of the LP relaxation at the current node. The aim of this approach is to drive the incumbent objective value $c^T z$ towards the objective value of the LP relaxation.

More formally, let z be the incumbent and z^* be the optimal solution of the LP relaxation. The set

$$E := \{j \in \mathcal{I} : z_j = z_j^*\} \tag{1.40}$$

contains all variables which have the same solution value in both the incumbent solution and the optimal solution of the LP relaxation. The created sub-MIP P_{RINS} is then defined as

$$\begin{aligned}
\text{Minimize} \quad & c^T x && && (1.41)\\
\text{s. t.} \quad & Ax \le b && && (1.42)\\
& l \le x \le h && && (1.43)\\
& x_j = z_j && \text{for } j \in E && (1.44)\\
& x_j \in \mathbb{Z} && \text{for } j \in \mathcal{I} && (1.45)\\
& x_j \in \mathbb{R} && \text{for } j \in \mathcal{C} && (1.46)
\end{aligned}$$

This approach was originally suggested in [DRL04] and implemented in SCIP by Berthold [Ber06]. The SCIP implementation is not activated by default.

1.4.5 Propagation heuristics

The presence of an upper bound $\bar{c}$ to the objective value $c^T z_{\text{opt}}$ of an optimal solution z_{opt} can help in pruning regions of the branch-and-bound tree. Another way to reduce the size of the tree consists in *domain propagation routines*. The knowledge of a local bound change of a variable during the branch-and-bound process is used to derive domain reductions for other variables to maintain the *consistency* [Apt03] at this particular node of the tree.

Propagation heuristics are primal heuristics which explore an auxiliary branch-and-bound tree in which they only use propagation routines and fixations. In particular, propagation heuristics do not solve an LP during their search.

Example 1 presents a MIP to which domain propagation can be applied.

Example 1. *Let P be a MIP problem containing the following row:*

$$1x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 \le 5 \tag{1.47}$$

with binary variables $x_j \in \{0, 1\}$, $1 \le j \le 5$. Assume that the branch-and-bound process has fixed x_2 to 1 locally. This information can be used directly to fix the variables x_4 and x_5 to the value of 0. Otherwise the sum of the coefficients would exceed the right hand side of (1.47).

Apart from domain propagation, *constraint propagation* for MIP is usually performed by updating row actitivities of an obtained solution. In that sense, the one opt 1.4.4 is a propagation heuristics because the feasibility of each of their variable shifts is propagated by keeping the LP-row activities up-to-date. The newly developed Shift and Propagate heuristic 4 will turn out to be a pure propagation heuristic.

Propagation heuristics in our sense have also already been referred to as *probing heuristics* by Achterberg [Ach07]. SCIP comes with an integrated propagation framework which can be used to expand a local branch-and-bound tree and apply propagation and backtracking techniques to derive feasible solutions. The propagation algorithms usually process all variables more than once whenever new fixations justify another *round*. This behaviour can be individually limited so as to reduce the running time of the propagation. The use of propagation routines locally at a branch-and-bound node can even have global effects: if the propagation detects an empty domain (a so-called *cutoff*), a *conflict clause* might be created under certain conditions which contains information about conflicting variables. After its creation, a conflict clause is globally available to support the branchandbound process.

Chapter 2

ZI round - a new MIP rounding heuristic

In [Wal09], Wallace suggests a new rounding heuristic called ZI round and which can be regarded as an extension of the simple rounding heuristic of SCIP, see 1.4.1.

In this chapter, we want to revisit the ideas with main focus on the algorithm and the computational results we obtained with our implementation in SCIP. The idea, names and definitions are all taken from Wallace [Wal09]. The implementation of ZI round in SCIP was done by the author of this thesis.

2.1 Outline of the approach

The main idea of ZI round is based on the definition of the fractionality of a variable and of a solution. Recall from Definition 2 that the fractionality $f_j^{\min}$ of a variable $j \in \mathcal{I}$ is defined as the distance of the solution value z_j of the variable in the LP relaxation to its closer integer neighbor,

$$f_j^{\min} := \min\{z_j - \lfloor z_j \rfloor, \lceil z_j \rceil - z_j\} \tag{2.1}$$

and the *integer infeasibility* of z as the sum over all variable fractionalities,

$$f^{\min}(z) = \sum_{j \in \mathcal{I}} f_j^{\min}. \tag{2.2}$$

From the definition, it is evident that an LP-feasible solution z is IP-feasible if and only if its integer infeasibility $f^{\min}(z) = 0$. In [Wal09], the fractionality $f_j^{\min}$ is denoted by $\mathrm{ZI}(z_j)$ which gave birth to the name of the heuristic.

A *positive shift* $z_j + \delta x_j$ of a variable solution value by a shift value $\delta x_j > 0$ which maintains LP-feasibility has to keep all LP row slacks nonnegative, hence it must satisfy

$$\delta x_j \leq ub := \min_i \left\{ \frac{s_i}{a_{ij}}, a_{ij} > 0 \right\}. \tag{2.3}$$

Additionally, the variable upper bound h_j must be considered for IP-feasibility. Thus, a positive shift

$$\delta x_j \leq \mathrm{UB} := \min\{ub, h_j - z_j^*\} \tag{2.4}$$

Algorithm 2: ZI round algorithm

Input : a MIP instance P, the LP-feasible relaxation solution z^*

Output : an IP-feasible solution $\hat{z}^*$ of P or NULL

$\hat{z}^* := z^*$;
while $f^{\min}(\hat{z}^*) > 0$ *and further improvements are possible* **do**
  foreach $j \in F$ *with fractional solution value* **do**
    Calculate UB$(\hat{z}^*_j)$, $LB(\hat{z}^*_j)$;
    if $f^{\min}(\hat{z}^*_j + UB(\hat{z}^*_j)) < f^{\min}(\hat{z}^*_j)$ **then**
      if $f^{\min}(\hat{z}^*_j + UB(\hat{z}^*_j)) < f^{\min}(\hat{z}^*_j - LB(\hat{z}^*_j))$ **then**
        $\hat{z}^*_j \leftarrow \hat{z}^*_j + UB(\hat{z}^*_j)$;
      else if $f^{\min}(\hat{z}^*_j + UB(\hat{z}^*_j)) > f^{\min}(\hat{z}^*_j - LB(\hat{z}^*_j))$ **then**
        $\hat{z}^*_j \leftarrow \hat{z}^*_j - LB(\hat{z}^*_j)$;
      else Shift $\hat{z}^*_j$ in the sense of the objective function ;
    end
    else if $f^{\min}(\hat{z}^*_j - LB(\hat{z}^*_j)) < f^{\min}(\hat{z}^*_j)$ **then**
      $\hat{z}^*_j \leftarrow \hat{z}^*_j - LB(\hat{z}^*_j)$;
    end
  end
end
if $f^{\min}(\hat{z}^*) = 0$ **then**
  return $\hat{z}^*$;
else return NULL;

is restricted by the row slacks and the variable upper bound. Similarly, a *negative shift* $z^*_j - \delta x_j$ must satisfy

$$\delta x_j \leq \mathrm{LB} := \min\{lb, z^*_j - l_j\} \tag{2.5}$$

where lb is defined as

$$lb := \min_i \left\{ \frac{-s_i}{a_{ij}}, a_{ij} < 0 \right\}. \tag{2.6}$$

The ZI round heuristic iteratively reduces the integer infeasibility of the LP relaxation solution z^* by searching for variables with fractional solution value $z^*_j \notin \mathbb{Z}$ which can be shifted satisfying the conditions above such that the new solution value $\hat{z}^*_j$ has lower fractionality $\hat{f}^{\min}_j < f^{\min}_j$. A shifting step is executed whenever there is a shifting direction, i.e., positive or negative, that reduces the fractionality. If both directions are possible, the direction is preferred which reduces the fractionality of this variable most. In the case that both directions reduce the fractionality by the same amount, we use the direction which improves the objective function. This strategy is summarized in Algorithm 2.

The presented algorithm can be regarded as an extension of the simple rounding algorithm (see Section 1.4.1). Since the simple rounding performs a rounding if a fractional variable $j \in \mathcal{I}$ has no up locks, i.e., if $a_{ij} \leq 0$ for all i, or no down locks, the corresponding values ub or lb equal ∞ by definition. This means that the variable is not locked by an LP row in at least one direction and can be trivially rounded by the ZI round heuristic. Hence, the set of solutions which ZI round finds is a superset of the solutions of simple rounding. Besides, in practically relevant MIPs a variable has more locks in the direction which improves the objective function. Simple rounding will always round in the direction which deteriorates the objective function value in that case. If both heuristics find a solution, the solutions of ZI round have the same or a better objective value. In the following section, we will try to verify this result from [Wal09].

Since a change, i.e., a shifting or rounding, of one solution value alters the slacks of the involved rows, it might enable us to change another solution value which could previously not be changed. Due to this consideration, it seems reasonable to iterate over the variables several times rather than only one time. This terminates when the solution $\hat{z}^*$ is IP-feasible or no further shifting is possible. In the following section, we will analyze in how far several loops improve the capability of the heuristic to find feasible solutions.

So far, the presented algorithm does not consider equations. Since equations are always fulfilled with a slack of zero by LP-feasible solutions, the calculated LB, UB in line 4 of Algorithm 2 are also zero for all candidate variables in that equation, hence, there is no shifting possible. Wallace suggests the following solution to this problem: whenever the problem contains an equation with at least one candidate variable, we search for an unfixed continuous variable $j \in \mathcal{C}$ which only appears in this particular row. Such a variable can be considered as slack variable. Note that in our notation of a MIP an equation is characterized by two matrix rows i,i' where $a_{i'j} = -a_{ij}$ for all j and $b_{i'} = -b_i$. Once a continuous variable and solution value $z_j^* \in [l_j, h_j] \subseteq \mathbb{R}$ with its (single) nonzero coefficient a_{ij} for row i is found, we can compute the slack s_i in the following way:

$$s_i := \begin{cases} a_{ij} \cdot (z_j^* - l_j),\ a_{ij} > 0 \\ -a_{ij} \cdot (h_j - z_j^*),\ a_{ij} < 0. \end{cases} \tag{2.7}$$

Such a slack variable can now be updated as follows: let δ_i be the total increase of the row activity caused by a shift of an integer variable. W. l. o. g., we assume $\delta_i > 0$, otherwise we will consider the second row i'. The resulting violation can be repaired by shifting the slack variable by $\delta x_j = \frac{-\delta_i}{a_{ij}}$. The new row slack $\hat{s}_i$ is computed as

$$\begin{aligned} \hat{s}_i &= a_{ij} \cdot (z_j^* + \delta x_j - l_j) && (2.8)\\ &= a_{ij} \cdot (z_j^* - l_j) + a_{ij} \cdot \delta x_j && (2.9)\\ &= s_i - \delta_i, \quad \text{if } a_{ij} > 0, && (2.10)\\ \hat{s}_i &= -a_{ij} \cdot (h_j - z_j^* - \delta x_j) && (2.11)\\ &= -a_{ij} \cdot (h_j - z_j^*) + a_{ij} \cdot \delta x_j && (2.12)\\ &= s_i - \delta_i \quad \text{also, if } a_{ij} < 0, && (2.13) \end{aligned}$$

Therefore, the heuristic has to keep track of all equations which contain at least one candidate variable, the specific slack variable, and the corresponding matrix coefficient in that equation. This additional computational effort is compensated by the higher ability of the heuristic to find solutions and the quick termination before starting the main algorithm.

2.2 Computational results in SCIP

ZI round has been introduced as an extension of simple rounding which is a straightforward algorithm to obtain IP-feasible solutions from LP-feasible ones. We want to compare in how far our implementation in SCIP is competitive with the already included simple rounding heuristic. Another interesting insight into the ZI round heuristic is whether its ability of finding solutions can be improved by increasing the number of iterations over the fractional variables, see Algorithm 2.

We will compare the two heuristics in two experiments: the first experiment is a direct comparison of the two heuristics called as the only heuristic directly after the initial root LP has been solved. We want to consider the number of IP-feasible solutions found as one measure for our comparison, since our main goal is to find feasible solutions for a given problem. We also want to take into account the required time of each heuristic. From the obeservations in the previous section, we expect ZI round to find at least as many solutions as simple rounding. Another result of Wallace is that both heuristics require the same time to find their solutions. We do not expect a similar result for our implementation of ZI round because computing the bounds UB, LB for a variable requires an iteration over the m_j LP rows of that variable, whereas the number of locks of the variable is computed dynamically during the solving process within SCIP and can be obtained in constant time by the simple rounding heuristic. We want to give an answer to the question whether the additional computational effort of ZI round is justified by the number and the quality of the obtained solutions.

We also want to delimit the maximum number of loops to a reasonable number. A good choice for the parameter `maxnloops` will find many solutions but will not significantly increase the running time compared to the quickest setting which executes only one loop. Hence, we will compare the different parameter settings by comparing the ratio of the number of found solutions over the sum of time spent on all problems.

For both heuristics, we disabled all other primal heuristics and delimited the number of solving nodes to 1. Moreover, we disabled the separation routines of SCIP to avoid a resolving of the LP. We set no limit to the maximum number of loops of ZI round.

With no further limitation to the number of loops, the ZI round heuristic succeeded in 45 of 162 cases whereas simple rounding found a solution on 35 instances or 29% less than ZI round. The test revealed that most of the solutions, namely 38 of the 45 or 84% of the solutions found by the heuristic were found after the first loop. Three more or 91% in total were found after the first two loops. The maximum number of loops, 11, was taken on the instance `sp97ar`. The result of Wallace could be confirmed that the solutions of ZI round have the same or a better objective value than those found by simple rounding. A strictly better solution was obtained in 13 out of 35 instances for which both heuristics succeeded. The results of this experiment are briefly summarized in Table 2.1. We also provide a table in all detail in Table 6.3, where we present the 45 instances for which ZI round was successful.

Regarding the solution quality and number, ZI round clearly outperforms simple rounding. The running time aspect, however, could not be clearly investigated in this single experiment. The required time by the heuristic never exceeded 20ms of CPU clock time during this experiment.

In the second test, we set a time limit of 30 min and no node limit, turned off simple

Loops	1	2	3	4	5	6 – 10	11	>11	$\sum$
Sols	38	3		2	1		1		45

Table 2.1: The benefit of multiple ZI round main loops given by the number of solutions found after the respective number of loops

rounding and set all other SCIP parameters to their default value. We will check the ZI round performance for four different parameter settings of the maxroundingloops parameter. In a fifth run, we will compare the competitivity of the best parameter setting against the SCIP default setting where ZI round is disabled. We excluded four problems from our test set, namely `markshare1_1` and `lrn` because they are known to be numerically unstable, and `nsrand-ipx` and `swath` because their solving process was cancelled prematurely due to insufficient memory.

The results of this second experimented are summarized in Table 2.2. Results for every loop limitating setting of ZI round are given in the respective column. The last column shows the results when ZI round was disabled. We see that ZI round is able to find solutions for at most 53 instances when no loop limitation is set. The total time spent on the execution of the heuristic increases by setting no loop limit. Delimitting the number of loops to 5 only slightly decreases the number of found solutions from 9283 to 9263 found solutions in comparison to no loop limit. The quickest parameter setting w. r. t. to the total heuristic runtime is the limitation to a single loop. This reduces the heuristic capability of finding solutions to 8800. This is a loss of 483 or 5 % of the total number of solutions in comparison to not setting a loop limit and a decrease of 6 in the number of instances on which solutions are found. Setting the loop limit to 5 results in the best quotient time/solutions. This line of the table also gives the impression that the heuristic is indeed fast since the time spent on one solution is about $2 \cdot 10^{-2}$ s on average. For the total time, the geometric mean time, the total number of nodes, and the geom. mean of solving nodes we only consider instances which could be solved by all runs within the time limit of 30 min.

Delimitting the number of loops to 1 results in the smallest total time spent on all instances, the standard SCIP setting disabling ZI round scores last in this respect. The default SCIP setting, however, wins in the number of nodes spent on all (solved) instances and both geometric means.

A detailed insight into the second experiment is presented in Table 6.4. To the left we present the results of a testrun with no ZI round loop limitation. The columns Time/sec and the left Nodes column refer to this test. In the column ZI-1 the number of solutions found by ZI round is given and can be compared to the number of solutions which ZI round finds when the number loops is restricted to at most 1, 2 or 5, respectively. In the remaining two columns at the right side of the table we present the results of the default testrun with simple rounding enabled instead of ZI round.

ZI round is an alternative to the simple rounding algorithm. It clearly outperformed simple rounding in our first experiment. The second experiment, however, has revealed that using ZI round has almost no impact onto the solving process of SCIP. A reason for this can be found in [Ach07]: a single heuristics like ZI round can only be exspected to have little impact on the entire benchmark set. Disabling an entire class of heuristics, e. g., rounding heuristics, would result in a deterioration of the solving process. Hence, ZI round is integrated in SCIP, because it has proven to be beneficial

Result	No limit	1 loop	2 loops	5 loops	No ZI round
ZI Solutions	9 283	8 800	8 799	9 263	
ZI Time/sec.	$2.31 \cdot 10^2$	$2.13 \cdot 10^2$	$2.17 \cdot 10^2$	$2.18 \cdot 10^2$	0
Time/Solution	$2.49 \cdot 10^{-2}$	$2.42 \cdot 10^{-2}$	$2.47 \cdot 10^{-2}$	$2.35 \cdot 10^{-2}$	0
ZI Problems	53	47	47	51	
Total Time/sec.	$2.11 \cdot 10^4$	$2.10 \cdot 10^4$	$2.10 \cdot 10^4$	$2.11 \cdot 10^4$	$2.12 \cdot 10^4$
Geom Mean. Time	$1.70 \cdot 10^1$	$1.69 \cdot 10^1$	$1.69 \cdot 10^1$	$1.70 \cdot 10^1$	$1.68 \cdot 10^1$
Total Nodes	45 087 798	45 048 701	45 048 701	45 037 986	44 639 978
Geom Mean. Nodes	$5.00 \cdot 10^2$	$4.99 \cdot 10^2$	$4.99 \cdot 10^2$	$4.99 \cdot 10^2$	$4.98 \cdot 10^2$

Table 2.2: Summary of the second ZI round experiment

during the root node experiment.

Chapter 3

A 2-opt heuristic for MIP

In this chapter we present the main ideas of a 2-opt heuristic for MIP. 2-opt is an improvement heuristic which does not rely on solving a sub-MIP. We will give an overview of the implementation and present computational results in the respective sections.

3.1 The k-opt heuristic for the TSP

Given a set of n cities with "travelling" cost coefficients $c_{i,j}$ for all $i \neq j \in \{1, \ldots, n\}$ the "cost" or "distance" from i to j, find a tour through all cities of minimal cost. The so called *Travelling Salesman Problem (TSP)* is one of the best-known among the various problems of combinatorial optimization and known to be $\mathcal{NP}$-hard. The TSP is important in many practical issues, e.g., steel production. Available heuristics for the TSP comprise methods of generating a start solution or improving a given solution. Amongst the latter ones, the *k-opt heuristic* with a fixed integer $k \geq 2$ is known to produce high quality solutions. The basic algorithm as, e.g., presented by Korte and Vygen in [KV02] is described in Algorithm 3.

Algorithm 3: k-opt algorithm

Input : An instance (K_n, c) of the *TSP*
Output : A tour

1 Let T be any tour, $E(T)$ its respective edge set ;
2 Let $\mathcal{S}$ be the family of k-element subsets of $E(T)$, the *edge set* of T ;
3 **foreach** $S \in \mathcal{S}$ *and all tours* T' *with* $E(T') \supseteq E(T) \setminus S$ **do**
4   **if** $c(T') < c(T)$ **then**
5         $T \leftarrow T'$;
6   `go to` 2 ;
7   **end**
8 **end**

The k-opt heuristic is a local search. The restriction to a fixed k garantees a polynomial size of the investigated neighborhood.

The tour returned by the k-opt algorithm is never worse than the initial tour due to the condition in line 4, yet it is not necessarily optimal unless $k = n$, for a counterexample we refer to [KV02] again. If a tour T cannot be improved further by the k-opt algorithm, we call T *k-opt* in order to distinguish between an *optimal* and a k-opt solution.

Further development based on this idea yielded the *chained Lin-Kernighan* heuristics. An advanced prototype of this family of heuristics is described by Applegate, Bixby, Chvátal and Cook in their computational study of the TSP [ABCC06, ch. 15]. They present and compare their implementation of chained Lin-Kernighan within the *Concorde* TSP solver using various data structures. For further information about Concorde, see [con].

3.2 Implementational details of 2-opt for MIP

In this section we want to adapt the combinatorial idea of k-opt to MIP. For $k = 1$ (which does not make much sense in the TSP context), the resulting *one-opt* heuristic has already been already available in SCIP, see 1.4.4. We will focus on the choice of $k = 2$ and consider pairs of variables to be set to new values at a time with the goal of improving the objective function while maintaining the IP-feasibility of the solution.

Algorithm 4: 2-opt for MIP

Input : a MIP P with $|\mathcal{B} \cup \mathcal{I}| \geq 2$, an IP-feasible solution z of P
Output : a solution $\hat{z}$ of P with improved objective $c^T\hat{z} < c^T z$, or **NULL**, if z is 2-opt

1 $Q \leftarrow \emptyset$;
2 Let $\mathcal{V} \subseteq \{(i,j) | \, i \neq j\} \subseteq \mathcal{I} \times \mathcal{I}$ be a family of pairs of integer variables ;
3 **foreach** $V = (i,j) \in \mathcal{V}$ **do**
4   Determine shift values $(\delta x_i, \delta x_j)$ s. t. $\varsigma_{ij} := c_i \cdot \delta x_i + c_j \cdot \delta x_j < 0$;
5   **if** $z_i + \delta x_i, z_j + \delta x_j$ *is feasible for* P **then**
6         $Q \leftarrow Q \cup (\delta x_i, \delta x_j, \varsigma_{ij})$;
7   **end**
8 **end**
9 **if** $Q = \emptyset$ **then return NULL**;
10 Sort the elements in Q in nondecreasing order of ς_{ij} ;
11 $\hat{z} \leftarrow z$;
12 **foreach** $(\delta x_i, \delta x_j, \varsigma_{ij}) \in Q$ **do**
13   **if** $\hat{z}_i + \delta x_i, \hat{z}_j + \delta x_j$ *is feasible for* P **then**
14         $\hat{z}_i \leftarrow \hat{z}_i + \delta x_i$;
15         $\hat{z}_j \leftarrow \hat{z}_j + \delta x_j$;
16   Update activities $a^T\hat{z}$ of the LP rows of P ;
17   **end**
18 **end**
19 **return** $\hat{z}$;

Example 2. *Let P be a MIP and z be IP-feasible for P. Let* (3.1) *be an equation of P,*

$$x_1 + x_2 = 1, \tag{3.1}$$

and $1, 2 \in \mathcal{B}$ be binary variables with solution values $z_1 = 1, z_2 = 0$. The one-opt heuristic fails in this situation to alter either of the two solution values because (3.1) *would be violated. 2-opt will flip x_1, x_2 to their opposite values and thus create a new solution.*

Algorithm 4 shows the ingredients of the approach. In analogy to the combinatorial expression, we will call a solution z *2-opt*, if it cannot be improved by Algorithm 4.

Algorithm 4 is generic in the sense that it contains two degrees of freedom: we will specify in the following sections how an appropriate subset $\mathcal{V} \subseteq \mathcal{I} \times \mathcal{I}$ in line 2 of Algorithm 4 is selected and how shift values $(\delta x_i, \delta x_j)$ in line 4 are determined. A first implementation might compare all integer variables to each other which yields an algorithm with *quadratic* running time $\mathcal{O}(|\mathcal{I}|^2)$ in the number of integer variables.

We will show that the heuristic can be significantly fastened by further restricting the search domain $\mathcal{V}$. Our final approach reflects the structure of the problem by rating the percentage of shared constraints between two variables, in order to reduce running time and preserve its efficiency in finding solutions.

3.2.1 Choosing pairs

In this section, we present an idea how promising variable pairs for the 2-opt heuristic can be chosen quicker. As mentioned above, an iteration over all possible pairs of variables is not desirable for running time reasons. The following example will motivate a more restricted approach.

Example 3. *Let P be a MIP with $\mathcal{I} = \{1, 2, 3, 4\}$ four integer variables and problem matrix A with*

$$A = \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} \color{red}{1} & 0 & \color{red}{1} & 0 \\ 0 & 1 & 0 & 1 \\ \color{red}{1} & 0 & \color{red}{1} & 0 \\ \color{blue}{1} & 0 & 0 & \color{blue}{1} \end{pmatrix} \end{array}. \tag{3.2}$$

We notice that variables 1 and 2 have no common row, i. e., a row for which both variables have a nonzero coefficient. A pairwise alteration of 1 and 2 would have no advantage over a one-opt approach because the variables do not share a single row of P. However, variable 1 shares two rows with variable 3 and another row with variable 4. The corresponding coefficients are highlighted in red and blue, respectively. The 2-opt heuristic will focus on such variables which share rows.

Example 3 motivates the definition of a more restrictive neighborhood of a variable to select pair candidates from.

Definition 4. *Let P be a MIP and $\gamma \in [0, 1]$ be a fixed* matching rate*. For a variable $j \in \mathcal{I}$ with nonzero coefficients A_j, we define the γ-*neighborhood

$$\Gamma_j^\gamma = \left\{ i \in \mathcal{I} \setminus \{j\} \middle| \frac{|A_j \cap A_i|}{|A_j|} \geq \gamma \text{ or } \frac{|A_j \cap A_i|}{|A_i|} \geq \gamma \right\} \tag{3.3}$$

to be all integer variables of P which share a certain ratio γ of rows with j. From the symmetry of the definition it holds

$$i \in \Gamma_j^\gamma \Leftrightarrow j \in \Gamma_i^\gamma. \tag{3.4}$$

For a given variable j, We consider its γ-neighborhood Γ_j^γ the most promising variable set to choose from for 2-opt, because it takes the structure of a given problem P into account. Variables which share (a proper ratio of) constraints like (3.1) are covered by Definition 4.

In Example 3, for $\gamma = 0.6$, the γ-neighborhood of variable 1 $\Gamma_1^{0.6} = \{3\}$ contains only variable 3. Choosing $\gamma \leq 0.5$, $\Gamma_1^\gamma = \{3, 4\}$. According to this concept, $\{2\} \notin \Gamma_1^\gamma$ unless $\gamma = 0$. We easily derive the following facts from Definition 4:

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \end{array} \qquad \begin{array}{cccc} x_1 & x_3 & x_4 & x_2 \\ \end{array}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \curvearrowright \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Figure 3.1: The variables from Example 3 before and after the lexicographical sortation of their column vectors

Lemma 1. *Let P be a MIP and $j \in \mathcal{I}$ be an integer variable. It holds*

$$\gamma = 0 \Rightarrow \ \Gamma_j^\gamma = \mathcal{I} \tag{3.5}$$
$$\gamma' \leq \gamma \Rightarrow \ \Gamma_j^\gamma \subseteq \Gamma_j^{\gamma'}. \tag{3.6}$$

For variables $(i, j) \in \mathcal{I} \times \mathcal{I}$, the detection whether $j \in \Gamma_i^\gamma$ takes $|A_i| + |A_j| = \mathcal{O}(\max\{|A_i|, |A_j|\})$ comparisons in the worst case. For reasons of symmetry we only need to consider pairs (i, j) with $i < j$ This yields an effort of

$$\sum_{i \in \mathcal{I}} \sum_{i < j \in \mathcal{I}} |A_i| + |A_j| = \mathcal{O}(|\mathcal{I}|^2 \cdot \max_{j \in \mathcal{I}}\{|A_j|\}) \tag{3.7}$$

comparisons in the worst case. The required memory for storing the neighborhood information for each variable will be $\mathcal{O}(|\mathcal{I}|)$ in the worst case. This sums up to $\mathcal{O}(|\mathcal{I}|^2)$ quadratic memory requirement w. r. t. the number of integer variables.

Both the running time and memory requirement can be improved by sorting the integer variable columns $A_j, j \in \mathcal{I}$ lexicographically. For the sorting, the vectors are interpreted as $\{0, 1\}^m$ vectors with 1 for each nonzero column entry. An example is given in Figure 3.1. After the sorting is done, a neighborhood structure can be detected by a single iteration over the integer variables as shown in Algorithm 5.

The sets β_j in Algorithm 5 are called *blocks*. The advantage of the sortation is that variable pairs (i, j) are now memorized implicitly with an array which consists of the first and the last variable index of the block β_j.

Lemma 2. *Algorithm 5 has a total running time of*

$$\mathcal{O}(|\mathcal{I}| \log |\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|). \tag{3.8}$$

Proof. Sorting takes $\mathcal{O}(|\mathcal{I}| \log |\mathcal{I}|)$ running time for constant time comparisons in $\mathcal{O}(1)$. A lexicographical comparison of two columns A_i, A_j requires $\mathcal{O}(\max\{A_i, A_j\})$ in the worst case. The lexicographical sortation can thus be implemented with a total effort of

$$\mathcal{O}(|\mathcal{I}| \log |\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|). \tag{3.9}$$

In the iteration 4, every variable $j \in \mathcal{I}$ is considered once during the loop and checked if it is an element of the current master neighborhood $\Gamma_{\text{master}}^\gamma$. The check requires $\mathcal{O}(\max\{|A_{\text{master}}|, |A_j|\})$. The running time of the entire loop is now

$$\mathcal{O}(|\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|) \tag{3.10}$$

Algorithm 5: procedure to choose 2-opt pairs

Input : A MIP P with problem matrix A, $\gamma \in [0, 1]$

Output : A subset $\mathcal{V} \subseteq \mathcal{I} \times \mathcal{I}$ for line 2 of Algorithm 4

1 Set $\mathcal{V} \leftarrow \emptyset$;
2 Set **master** $\leftarrow$ **NULL**;
3 Let A_j^{01} be the column vector of $j \in \mathcal{I}$ after replacing all nonzero entries by 1 ;
4 **foreach** $j \in \mathcal{I}$ *in lexicographical order of* A_j^{01} **do**
5   **if** **master** $=$ **NULL** **then**
6   **master** $\leftarrow j$;
7   initialize *master block* $\beta_{\mathbf{master}} \leftarrow \emptyset$;
8   **continue**;
9   **else**
10   **if** $j \in \Gamma^{\gamma}_{\mathbf{master}}$ **then**
11             $\beta_{\mathbf{master}} \leftarrow \beta_{\mathbf{master}} \cup \{j\}$;
12   **else**
13             $\mathcal{V} \leftarrow \mathcal{V} \cup (\beta_{\mathbf{master}} \times \beta_{\mathbf{master}})$;
14   **master** $\leftarrow j$;
15   reset $\beta_{\mathbf{master}} \leftarrow \emptyset$;
16   **end**
17   **end**
18 **end**
19 **return** $\mathcal{V}$;

comparisons in the worst case. Since we have a sorted array of integer variables at hand, we only need to keep track of the indices of the first and the last variable of the current block β_{master} to describe $\mathcal{V}$. Since the total number of blocks β_j is always smaller or equal to the number of integer variables $\mathcal{I}$, we have a linear memory requirement for Algorithm 5. Hence, the number of memory accesses is dominated by the number of comparisons during loop 4 and not considered in the asymptotic notation. The sum of (3.9) and (3.10) is the running time of Algorithm 5. Hence, we get

$$\mathcal{O}(|\mathcal{I}| \log |\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|) + \mathcal{O}(|\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|) = \mathcal{O}(|\mathcal{I}| \log |\mathcal{I}| \cdot \max_{j \in \mathcal{I}} |A_j|) \tag{3.11}$$

as overall running time. □

Note that with a choice $\mathcal{V}$ as in Algorithm 5, only a subset of pairs of the set

$$\bigcup_{j \in \mathcal{I}} \{(i, j) | i \in \Gamma_j^{\gamma}\} \tag{3.12}$$

is considered in order to reduce running time and memory requirements of the heuristic.

Furthermore we treat binary and integer variables separately in the current implementation of 2-opt in SCIP, i. e., two families of pairs are created by Algorithm 2, $\mathcal{V}_{\mathcal{B}}$ and $\mathcal{V}_{\mathcal{I}}$.

3.2.2 Determining new solution values

The 2-opt construction of TSP tours is intuitive because deleting two edges from a tour offers a single possibility of recombination. In MIP problems, however, it is not clear a priori how to alter the solution values of variable pairs and maintain the IP-feasibility.

Once a promising pair (i, j) of variables of a given solution z has been chosen, a feasible alteration of their solution values must be determined. We decided to consider $1:1$-shiftings, i. e., both variables are shifted by the same value

$$|\delta x_i| = |\delta x_j|. \tag{3.13}$$

For the shifting directions there are at most four possible combinations. At most two of them can improve the objective function, one for the same direction, i. e., either both upwards or both downwards, and one for opposing directions. In the following, we describe how to choose a shift value for opposing directions, $\delta x_i = -\delta x_j$. This can be straightforward adapted to the case of equal shifting directions.

Let z be IP-feasible for P z_i, z_j be the solution values and c_i and c_j be the objective coefficients of i and j, respectively. If $c_i < c_j$ we want to shift i upwards by $\delta x_i = -\delta x_j > 0$, because a solution obtained by setting $\hat{z}_i := z_i + \delta x_i$ and hence $\hat{z}_j := z_j - \delta x_i$ yields a new solution $\hat{z}$ with objective value

$$\begin{aligned} c^T \hat{z} &= c^T z + c_i \cdot \delta x_i - c_j \cdot \delta x_i && (3.14)\\ &= c^T z + \delta x_i (c_i - c_j) && (3.15)\\ &< c^T z, && (3.16) \end{aligned}$$

and if $c_i > c_j$ we want to shift i downwards. W. l. o. g., let i denote the variable we want to shift upwards.

Definition 5. *Let P be a MIP, (i, j) be a pair of variables, r the r-th row of A. We call r* affected *by i, j, if i or j appear in i with a nonzero coefficient, i. e.,*

$$r \in A_i \cup A_j. \tag{3.17}$$

In order to maintain feasibility, the choice of δx_i depends on the remaining *slack* s_r to the right hand side b_r of every affected LP row r.

Lemma 3. *Let P be a MIP, (i, j) be a pair of variables and z IP-feasible for P. Define*

$$\delta_r := \begin{cases} \lfloor \frac{s_r}{a_{ri} - a_{rj}} \rfloor, & a_{ri} - a_{rj} > 0, r \in A_i \cup A_j \\ \infty, & else \end{cases} \tag{3.18}$$

and $\delta x_i = \delta := \min \delta_r$. δ is the maximum nonnegative integer by which (i, j) can be shifted without losing feasibility.

Proof. W. l. o. g., we can assume that $\delta < \infty$ because otherwise P is unbounded. δ is always nonnegative because all δ_r are nonnegative. For any affected row r, the difference $a_{ri} - a_{rj}$ is either nonpositive, which means that the slack s_r is not shrinked, or the slack is reduced by $(a_{ri} - a_{rj}) \cdot \delta$. The slack $\hat{s}_r$ is

$$
\begin{aligned}
\hat{s}_r &= s_r - (a_{ri} - a_{rj}) \cdot \delta && (3.19)\\
&\geq s_r - (a_{ri} - a_{rj}) \cdot \delta_r && (3.20)\\
&\geq s_r - (a_{ri} - a_{rj}) \lfloor \frac{s_r}{a_{ri} - a_{rj}} \rfloor && (3.21)\\
&\geq s_r - (a_{ri} - a_{rj}) \frac{s_r}{a_{ri} - a_{rj}} && (3.22)\\
&= 0 && (3.23)
\end{aligned}
$$

which means that row r remains feasible after the shift. It is *maximal*, because for any row r for which δ_r is minimal, a shift of $\delta + 1$ yields a slack

$$
\begin{aligned}
\hat{s}_r &= s_r - (a_{ri} - a_{rj})(\delta + 1) && (3.24)\\
&= s_r - (a_{ri} - a_{rj})(\delta_r + 1) && (3.25)\\
&= s_r - (a_{ri} - a_{rj})(\lfloor \frac{s_r}{a_{ri} - a_{rj}} \rfloor + 1) && (3.26)\\
&< s_r - (a_{ri} - a_{rj}) \frac{s_r}{a_{ri} - a_{rj}} && (3.27)\\
&= 0, && (3.28)
\end{aligned}
$$

and thus yields a violation since (3.27) is sharp. □

If $\delta = 0$ is zero, a shift is not within feasible range. Note that for an equation, like (3.1) in Example 2, the two LP rows r, r' in our notation contribute either $\delta_r = 0$ or $\delta'_r = 0$, if $a_{ri} \neq a_{rj}$, or $\delta_r = \delta'_r = \infty$ for equal coefficients $a_{ri} = a_{rj}$. Determining δ can be performed in

$$\mathcal{O}(|A_i \cup A_j|) = \mathcal{O}(\max\{|A_i|, |A_j|\}). \tag{3.29}$$

3.3 Computational results

After having introduced the two decisions to be made concerning variable selection and value selection in the previous sections, we test our implementation of 2-opt in SCIP. We will see that 2-opt in its general form presented in Algorithm 4 is indeed quite time consuming with an increasing number of variables. Furthermore, we will analyze the ideas to reduce the size of the search domain of the heuristic, as presented in the preceding section. It is an interesting question if our results obtained with $k = 2$ motivate further approaches for MIP similar to those of Applegate, Bixby, Chvátal, and Cook [ABCC06] for $k \geq 3$.

Our first aim is to examine the quality of 2-opt with a matching rate of $\gamma = 0.0$ in terms of the number of found solutions and the required time of the heuristic. Recall that setting $\gamma = 0.0$ will consider all possible pairs of binary variables and integer variables.

3.3.1 2-opt with $\gamma = 0.0$

The implementation of 2-opt can be adjusted to consider all pairs of binary/integer variables by setting the parameters `matchingrate` to 0.0 and `maxnslaves` to -1. We excluded the instances markshare1_1 and lrn because they are known to be numerically unstable, and swath and nsrand-ipx because both of them exceeded our memory limit prematurely. We ran the test at a time limit of 1800 sec per instance.

The results of the test can be seen in Table 3.1. We only present instances for which 2-opt was successful or required more than 10 % of the overall solving time.

2-opt found solutions on 25 out of 158 instances. On 3 problems, it finds the primal bound. We have a total of 32 timeouts or one more than obtained with 2-opt disabled. On 16 problems, more than 10 % of the overall solving time are spent on the 2-opt heuristic. For fast0507, 94 % of the solving time is spent on 2-opt. Particular focus lies on instances like neos6, nw04, fast0507, irp, t1717 for which 2-opt needs a lot of solving time but cannot find a new primal solution. The reason for the long running time of the heuristic is primarily a large number of discrete variables in the problem. In Table 3.2, we show instances for which 2-opt found solutions or was time consuming, i. e., took more than 10 % of the overall solving time. The problems have been sorted by the number of variables for a better overview. See also Figure 3.2 where the variable influence has been plotted.

The number of solutions is only an indicator of the capability of 2-opt to find good primal solutions. Recall that if, e. g., a solution of the LP relaxation was succesfully rounded to an IP-feasible one, the resulting primal-dual gap might already be considerably small. In such a case, it is unlikely for an improvement heuristic with a restricted search domain to contribute better solutions. Moreover, the order in which SCIP calls the heuristics plays a big role. SCIP calls heuristics in descending order of their priority parameter. For our tests, we assigned a smaller priority to 2-opt than to one-opt in accordance with our general expectation that 2-opt should still be able to improve solutions in the case that one-opt fails. This might decrease the number of possible solutions compared to calling the heuristics in a different order.

The running time of this version of the 2-opt heuristic remains problematic for practical issues. We can see in Figure 3.2 that the increase in the running time is approximately quadratic in terms of the increase in the number of variables. We also see problems of the same variable number differing in the running time of the heuristic. An explanation for that is the number of LP-rows of the problems which also has a strong influence on the solving time.

3.3.2 2-opt for $\gamma > 0$

In this section, we will investigate the influence of the matchingrate parameter. Recall that the variables are grouped into blocks if they share at least a ratio of γ of their LP-rows after a lexicographical sortation of the LP-columns. We tested the performance for $\gamma \in \{0.1, 0.25, 0.4, 0.6, 0.75, 0.95\}$. These values were chosen to reflect a good coverage of the possible values for the continuous parameter γ. Apart from setting the matchingrate γ to the specific value, we did not further restrict the number of candidates a single variable can be paired with from a common block by setting the parameter maxnslaves to -1.

Table 3.3 shows the influence of the matching rate parameter for the running time of the heuristic. The column 0.0 represents the running time of 2-opt for $\gamma = 0.0$. The next columns show the running time of the heuristic for specific matching rates on the set of instances where the full 2-opt heuristic was relatively slow or found solutions. The last column *Time(sec)* contains the default solving time of the problem with

Problem Name	2-opt Time/sec	Sols	Time/sec	Time limit
30:70:4_5:0_5:100	43.44	2	177.96	
30:70:4_5:0_95:98	45.20	3	141.55	
30:70:4_5:0_95:100	45.68	2	170.93	
a1c1s1	0.41	2	1 800.00	time
air04	24.39	0	71.79	
air05	28.37	0	51.98	
atlanta-ip	115.68	2	1 800.01	time
fast0507	1 795.49	0	1 922.87	time
fiber	0.09	1	1.04	
gt2	0.00	1	0.08	
harp2	1.92	3	244.08	
irp	274.83	0	322.73	
l152lav	1.71	0	3.98	
manna81	0.87	0	1.27	
markshare1	1.54	2	1 564.52	
markshare2	1.27	2	1 614.36	
markshare2_1	2.44	1	1 800.00	time
markshare4_0	0.22	1	124.19	
mitre	0.13	1	6.25	
mod010	0.36	0	1.41	
mod011	0.06	1	66.89	
mzzv11	16.06	2	299.40	
neos6	40.22	0	152.75	
ns1688347	0.21	1	511.02	
ns1671066	3.05	6	1 800.00	time
nw04	948.09	0	1 044.12	
p0201	0.05	3	0.84	
p0282	0.01	1	0.43	
p0548	0.03	2	0.23	
p2756	0.45	2	1.96	
pp08a	0.00	1	1.05	
pp08aCUTS	0.01	1	1.01	
qiu	0.01	1	60.88	
qnet1_o	0.34	0	2.05	
sp97ar	66.91	3	1 800.00	time
t1717	1 322.09	0	1 961.97	time

Table 3.1: Results of the first experiment of 2-opt with $\gamma = 0.0$

Problem Name	Nr. Variables	Time(sec)/call
markshare4_0	30	0.03
qiu	48	0.00
markshare1	50	0.19
markshare2_1	54	0.41
markshare2	60	0.07
pp08a	64	0.00
pp08aCUTS	64	0.01
p0282	65	0.01
mod011	96	0.01
gt2	173	0.00
a1c1s1	192	0.02
p0201	195	0.01
p0548	235	0.01
fiber	444	0.09
harp2	1 045	0.09
p2756	1 072	0.09
qnet1_o	1 330	0.05
ns1688347	1 590	0.07
l152lav	1 989	0.17
mod010	2 014	0.36
ns1671066	2 660	0.18
manna81	3 321	0.87
mitre	4 941	0.13
air05	6 117	2.58
air04	7 375	2.71
neos6	8 340	20.11
mzzv11	8 802	1.78
30:70:4_5:0_5:100	10 763	10.86
30:70:4_5:0_95:98	10 919	11.30
30:70:4_5:0_95:100	10 973	11.42
sp97ar	14 100	16.73
atlanta-ip	16 140	23.14
irp	20 315	14.46
nw04	61 369	316.03
fast0507	62 997	448.87
t1717	68 042	661.04

Table 3.2: The number of variables and the running time of 2-opt.

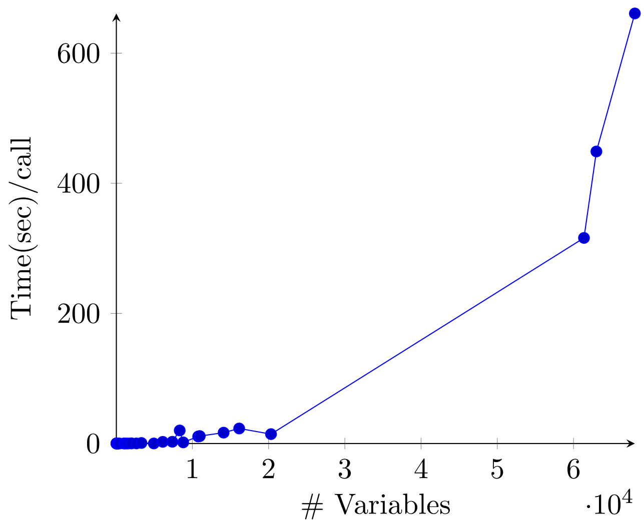


Figure 3.2: The time consumption of 2-opt for $\gamma = 0.0$ on the instances of the tables 3.1 and 3.2.

SCIP version 2.0.1 and default settings. We see that the heuristic could be extremely fastened on all instances for which the heuristic was time consuming for $\gamma = 0.0$. Table 3.3 illustrates the running time of our focussed set of problems and the possibility to achieve practically reasonable running times by adjusting the matchingrate.

Since every increase of the matchingrate shrinks the search domain of the heuristic, less solutions can be expected by increasing γ. The effects of the restrictive approach on the ability to find solutions are shown in Table 3.5. 2-opt found solutions on 25 instances with a matching rate $\gamma = 0.0$. The best matching rate $\gamma > 0.0$ in this test was $\gamma = 0.25$ which was still successful on 18 problems. For $\gamma > 0.25$, the number of found solutions decreases monotonously. The fewest solution number was achieved by the 0.95-matchingrate which could only contribute new solutions to 6 instances of the testset.

It is a surprising fact that the number of solutions as well as the number of problems for which a solution was found is not monotonously decreasing with an increasing matching rate. The reason for this behaviour lies in the selection of the neighborhood blocks. The block structure can be altered significantly by changing the matching rate γ s. t. variables which were grouped together before are now in different blocks. Such a behaviour can be tolerated for the greater efficiency.

We will compare the different 2-opt heuristic runs to the SCIP default settings by the following measure: Whenever a problem was solved by both settings within the time limit, we compare the required solving time. If only one setting hits the memory or time limit, this is worse. If both settings hit the time/memory limit, we compare the gap between the primal bounds of the settings to the optimal or best known solution. If both settings terminate with an equal gap, the better setting shall be the one which needs less time to find the primal bound.

Table 3.4 comprises the results of the experiment. The $T_{\min}$ column shows for each of the different settings on how many instances the corresponding solver reached the best evaluated solving time in case that at least one solver did not hit the time limit. The column $T < T_0$ compares each of the settings to SCIP default settings.

Problem Name	0.0	0.1	0.25	0.4	0.6	0.75	0.95	Time(sec)
30:70:4_5:0_5:100	43.44	0.12	0.13	0.05	0.04	0.03	0.02	133.65
30:70:4_5:0_95:98	45.20	0.13	0.10	0.08	0.04	0.05	0.01	96.04
30:70:4_5:0_95:100	45.68	0.15	0.11	0.08	0.05	0.05	0.04	124.44
a1c1s1	0.41	0.19	0.17	0.20	0.05	0.03	0.08	1 800.00
air04	24.39	24.39	0.84	0.13	0.05	0.01	0.01	46.96
air05	28.37	11.69	2.19	0.22	0.03	0.03	0.01	23.45
atlanta-ip	115.68	84.92	26.67	12.59	1.46	1.13	0.31	1 800.01
fast0507	1 795.49	35.00	4.53	1.54	0.56	0.36	0.23	254.74
fiber	0.09	0.00	0.01	0.00	0.00	0.00	0.00	0.93
gt2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
harp2	1.92	0.85	0.79	0.17	0.05	0.04	0.02	139.65
irp	274.83	48.51	15.68	9.93	3.41	0.57	0.18	47.03
l152lav	1.71	1.71	0.46	0.06	0.01	0.01	0.00	2.24
manna81	0.87	0.01	0.00	0.00	0.00	0.00	0.00	0.41
markshare1	1.54	1.55	1.60	1.56	1.83	1.80	1.85	1 461.44
markshare2	1.27	1.22	1.68	1.58	1.06	1.04	1.20	1 573.05
markshare2_1	2.44	2.25	2.61	2.09	1.94	2.31	2.44	1 800.00
markshare4_0	0.22	0.12	0.17	0.16	0.12	0.18	0.30	157.26
mitre	0.13	0.01	0.01	0.01	0.00	0.01	0.00	6.00
mod010	0.36	0.37	0.36	0.01	0.00	0.00	0.00	1.05
mod011	0.06	0.03	0.05	0.02	0.02	0.03	0.01	64.96
mzzv11	16.06	0.10	0.05	0.06	0.06	0.08	0.06	282.43
neos6	40.22	2.12	1.71	0.77	0.20	0.05	0.04	111.74
ns1688347	0.21	0.02	0.03	0.03	0.02	0.02	0.01	456.30
ns1671066	3.05	0.31	0.38	0.36	0.35	0.28	0.23	1 800.00
nw04	948.09	87.25	22.40	9.76	1.28	0.58	0.33	95.11
p0201	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.80
p0282	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.43
p0548	0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.21
p2756	0.45	0.07	0.02	0.04	0.02	0.02	0.02	1.83
pp08a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.04
pp08aCUTS	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.97
qiu	0.01	0.00	0.00	0.00	0.00	0.00	0.00	59.26
qnet1_o	0.34	0.00	0.00	0.00	0.00	0.00	0.00	1.70
sp97ar	66.91	5.14	1.72	1.61	1.00	0.47	0.21	1 800.01
t1717	1 322.09	13.04	2.00	1.28	0.69	0.49	0.41	1 800.03

Table 3.3: Running time of 2-opt for different parameter settings of the matching rate.

Test	$T_{\min}$	$T < T_0$	$T > T_0$	$\Delta < \Delta_0$	$\Delta > \Delta_0$	$t_{\min}$	$t < t_0$	$t > t_0$
0.0	15	21	72	5	4	2	0	19
0.1	18	26	60	4	4	3	2	18
0.25	23	29	60	3	5	0	0	20
0.4	25	36	61	4	4	0	0	20
0.6	18	21	68	3	4	2	1	20
0.75	19	25	63	3	3	0	2	20
0.95	15	23	68	3	3	2	3	19
def	54	0	0	0	0	19	0	0

Table 3.4: Summary of the comparison between different settings of the matching rate parameter and the default SCIP.

In case that all test runs hit either the time or memory limit, we focus on the gap between the best solution found so far by the solver. The column $\Delta < \Delta_0$ indicates on how many problems the produced gap was strictly better than the one by the default settings. In case of equal gaps, particularly a 0.0 % gap, we compare the solving times until the best solution was found. The column $t < t_0$ shows how often the respective solvers outperformed the default settings of SCIP. There are also instances for which the overall solving time, the time to the best solution or the gap are worse for the 2-opt matching rate settings compared to SCIP default. The respective number of instances are presented in the columns $T > T_0$, $t > t_0$ and $\Delta > \Delta_0$, resp. On a total of 133 problems, the different matching rate settings score better than the SCIP default solver on 21–36 instances. There are also slight time reductions considered. The leading 2-opt matching rate parameter is $\gamma = 0.4$ which is also the leading setting w. r. t. The best setting in this respect is the $\gamma = 0.0$ 2-opt setting with 5 gap reductions, followed by the matching rates 0.1 and 0.4 with 4 reductions each.

On the other hand, all 2-opt settings terminate the run with a worse gap for about the same number of instances. Comparing the required time until the best solution was found the 2-opt settings score better than SCIP default for 0–3 cases. In this respect the matching rate 0.95 scores best. The number of instances for which the solving time increases with 2-opt ranges from 60–72. $\gamma = 0.1, 0.25$ score best in this respect with 60 instances each for which the solving time increases. The worse matching rate is $\gamma = 0.0$ with 72 instances.

3.3.3 Conclusions

We introduced the 2-opt heuristic for MIP as a generalization of the combinatorial k-opt heuristic for the TSP for $k = 2$. In our experiments, we have seen that an approach which pairs all variables to each other is not favorable for MIP because the benefit in finding solutions is outweighed by a long running time with an increasing number of variables. The Γ-neighborhoods introduced in section 3.2.1 can reduce the running time of 2-opt to a practically reasonable amount. The results show that all parameter configurations of 2-opt are beneficial on a small set of instances for which we measured a gap or time-to-best reduction when the time limit was run. The heuristic is promising, if it is used in an aggressive setting of the solver.

Running time reducement can also be achieved by limiting the maximum number of candidates a variable is compared to. In our current implementation in SCIP,

Problem Name	0.0	0.1	0.25	0.4	0.6	0.75	0.95
30:70:4_5:0_5:100	2	2	2	1	1	1	
30:70:4_5:0_95:98	3	2	2	2	1	2	
30:70:4_5:0_95:100	2	2	2	2	1	1	
a1c1s1	2	1	1	1			
atlanta-ip	2						
fiber	1	1	1	1	1	1	1
gt2	1						
harp2	3	1	1	1			
markshare1	2	2	2	2	1	1	1
markshare2	2	2	2	2	2		
markshare2_1	1	1	1	1	1	1	1
markshare4_0	1	1	1	1	2	2	1
mitre	1						
mod011	1	1	1				
mzzv11	2						
ns1688347	1		1	1	1		
ns1671066	6	6	6	6	6	6	
p0201	3	3	3	3	3	3	
p0282	1	1	1	1			
p0548	2	2	2	2	2	2	3
p2756	2	2	2	2	2	3	
pp08a	1						
pp08aCUTS	1						
qiu	1						
sp97ar	3	4	2	4	5	4	3

Table 3.5: The capability to find solutions for specific matching rates of the 2-opt heuristic

one can do so by adjusting the parameter `maxnslaves` to a positive value. Future implementations could also feature an adaptive choice of the matching rate parameter. Given a time-consuming problem, the matching rate can be altered depending on the success of the heuristic and its time-consumption.

An implementation of the k-opt heuristic for $k \geq 3$ remains a topic for future research. Our tests for $k = 2$ underline the need for a sophisticated search strategy like our neighborhood blocks.

Chapter 4

The Shift and Propagate pre-root heuristic

Shift and Propagate (SaP) is a new start heuristic for MIP which aims at finding a feasible solution before the root node LP relaxation is solved. After a suitable problem transformation, the Shift and Propagate algorithm creates a small search tree in which one integer variable gets fixed at every node. The fixation is then propagated to reduce the domain of other variables. In contrast to diving heuristics, SaP does not rely on the solution of the LP relaxation. Besides, the heuristic does not solve LP relaxations at every node during the search. This chapter presents the steps of the algorithm in detail and discusses computational results on our test set. The Shift and Propagate implementation has been done by the author of this thesis and has been integrated as default plugin in SCIP since version 2.0.

4.1 Introduction

"The early bird catches the worm." is a saying which can be perfectly applied to solving MIP. Feasible solutions play a big role in the branch-and-bound search. The trivial heuristic (see 1.4) has been the only approach implemented in SCIP so far which finds feasible solutions before the root LP is solved. Four solutions are created which are often feasible in models of practical relevancy. In many of those cases, triviality of a solution and the quality of its objective value counteract each other. Hence a solution found by the trivial heuristic often has a weak objective value. Furthermore, no structural information about the problem and its constraints is used (except for the *lock solution*). Example 4 shows a small MIP problem for which none of the trivial solutions are feasible.

Example 4.

$$x_1 + x_2 + x_3 = 1 \tag{4.1}$$
$$-x_0 + x_1 \leq 0 \tag{4.2}$$
$$-x_0 + x_2 \leq 0 \tag{4.3}$$
$$-x_0 + x_3 \leq 0 \tag{4.4}$$
$$x_0, x_1, x_2, x_3 \in \{0, 1\}. \tag{4.5}$$

The problem consists of a set partitioning equation (4.1) *and three linear inequalities. The feasible solutions to this problem are given by* $x_0 = 1$ *and* $x_i = 1$*,* $x_j = 0$*,* $i \in \{1, \ldots, 3\}$*,* $j \in \{1, \ldots, 3\} \setminus \{i\}$*. Note that none of the four solutions obtained by the trivial heuristic are feasible. Nonetheless, starting with the zero solution, it is easy to find a feasible solution manually by noticing that setting the variable* x_0 *to a value of* 1 *provides the choice of setting one of the remaining* x_i*-variables to* 1*, as well. Thus, the shift of* x_0 *to* 1 *is mandatory to make constraint (4.1) feasible although the variable is not part of it.*

In contrast to the trivial heuristic, the SaP heuristic takes more structural information into account. The details of Shift and Propagate are subject to the next section.

4.2 Implementational details

Our main goal is to develop a pre-root heuristic which does not depend on a solution of the LP-relaxation as do, e. g., rounding or diving heuristics.

In this section, the *best shift* criterion how to set a solution value is discussed. Besides we will introduce a comparison method for variables to measure the *importance* of a variable for the overall problem. The best shift criterion is used by SaP to fix variables for propagation. This main procedure is preceded by a suitable *problem transformation* which is presented first.

4.2.1 Problem transformation

Another look at Example 4 reveals a number of properties of the problem:

1. the lower bounds of all variables are zero,
2. for each row $a^T x \leq b$ holds $\|(a^T)\|_\infty = 1$ with the *infinity-norm*

$$\|x\|_\infty = \max_{i=1}^{n} |x_i| \tag{4.6}$$

3. the problem is an IP, i. e., it contains no continuous variables.

The lower bound property 1 has the advantage that the zero solution satisfies all variable bound restrictions $l_j \leq x_j \leq h_j$. The (in-)feasibility of a row $a^T x \leq b$ w. r. t. the zero solution $z_0 \equiv 0$ depends on whether $b \geq 0$ or $b < 0$. Besides, the choice of a possible solution value is simplified because every possible solution value represents a shift of the variable solution value in positive direction. The advantage of the normalization property 2 is that we can compare the coefficients of different rows to measure the importance of a variable for the overall problem. Any MIP problem P can be transformed s. t. P has the properties 1 and 2 by the following transformations:

Lemma 4. *Given an integer variable* $j \in \mathcal{I}$ *with bounds* $l_j \leq x_j \leq h_j$ *with* $l_j, h_j \in \mathbb{Z}$*, we can substitute* x_j *by* $\bar{x}_j$

$$\bar{x}_j = \begin{cases} x_j - l_j, \ |l_j| \leq |h_j| \\ h_j - x_j, \ |h_j| < |l_j| \end{cases} \tag{4.7}$$

and obtain a variable with lower bound $\bar{l}_j = 0$*. Every row* $a_i^T x \le b_i$, $1 \le i \le m$ *with* $b_i \in \mathbb{R}$ *is transformed to* $\bar{a}_i^T x \le \bar{b}_i$ *with*

$$\bar{b}_i = \begin{cases} b_i - a_{ij} l_j, \ |l_j| \le |h_j| \\ b_i - a_{ij} h_j, \ |h_j| < |l_j| \end{cases} \tag{4.8}$$

$$\bar{a}_{ij} = \begin{cases} a_{ij}, \ |l_j| \le |h_j| \\ -a_{ij}, \ |h_j| < |l_j| \end{cases} \tag{4.9}$$

and we obtain an equivalent problem $\bar{P}$*.*

Lemma 5. *Let* z *be feasible for* $a^T z \le b$ *with* $\|a^T\|_\infty > 0$*. Then* z *is a feasible solution of the* normalized *constraint* $\frac{a^T z}{\|a^T\|_\infty} \le \frac{b}{\|a^T\|_\infty}$*.*

The transformations from Lemmas 4 and 5 require $\mathcal{O}(|A^{\neq 0}|)$ comparisons and assignments. The transformation in Lemma 4 is also possible for variables $j \in \mathcal{I}$ for which either $l_j = -\infty$ or $h_j = \infty$ but not both. In case that j is *free*, i. e., it has both a lower bound $l_j = -\infty$ and an upper bound $h_j = \infty$, it can be replaced by two integer variables $x_j \leftarrow x_j^+ - x_j^-$ with lower bound $l_{j^+} = l_{j^-} = 0$.

In praxis, many variables already have a lower bound of zero, e.g., binary variables. For those variables, the transformation suggested in Lemma 4 is the identity transformation and can thus be skipped.

In case of continuous variables in P, we can use these variables to enlarge our temporary search domain by a transformation which we call *relaxing by continuous variables.*

Definition 6. *Let* x_j, $j \in \mathcal{C}$ *be a variable of a MIP* P *with a continuous domain* $[l_j, h_j] \subseteq \tilde{\mathbb{R}}$ *and* $a_i^T x \le b_i$ *a row of* P*. Depending on the sign of the coefficient* a_{ij}*, we change the right hand side* b_i *to* $\bar{b}_i$ *and eliminate* j *from the row by setting*

$$\bar{b}_i \leftarrow \begin{cases} b_i - a_{ij} \cdot h_j, \ a_{ij} < 0 \\ b_i - a_{ij} \cdot l_j, \ a_{ij} \ge 0 \end{cases}$$

$$\bar{a}_{ij} \leftarrow 0$$

and call this operation eliminating j from the row $a_i^T x \le b_i$*. We call the obtained problem* $\bar{P}$ *the* IP-relaxation *of* P *after all continuous variables have been eliminated from their rows.*

Please observe that in Definition 6, the right hand side b_i can even be altered to $\bar{b}_i = \infty$. Such a relaxed row is redundant for $\bar{P}$. Another simplification achieved by this relaxation is the transformation of equation constraints into so-called ranged row constraints, which are hopefully easier to fulfill. The relaxing of $x_j \in [l_j, h_j]$ from an equation $a^T x = b$ with, w. l. o. g., $a_j > 0$, yields a constraint

$$b - a_j \cdot h_j \le \bar{a}^T x \le b - a_j \cdot l_j. \tag{4.10}$$

Another beneficial outcome of relaxing continuous variables is presented in Example 5.

Example 5. *Let P be a MIP with the constraints*

$$\begin{aligned} x_1 + x_2 + x_3 &\leq -1, \\ x_1, x_2 &\in \{0,1\}, \\ x_3 &\in [-2,2]. \end{aligned} \tag{4.11}$$

P contains two binary variables, 1 and 2, and one continuous variable 3. The zero solution z_0 violates (4.11) by 1. Our goal is to eliminate 3 from the row so that we can concentrate on the binary variables. Any feasible solution of x_1 and x_2 must preserve the changed inequality

$$x_1 + x_2 \leq -1 - x_3 \leq -1 - (-2) = 1 \tag{4.12}$$

because of the lower bound $l_3 = -2$ of variable 3. For the obtained IP-relaxation $\bar{P}$, z_0 is feasible.

The suggested transformations and observations from this chapter justify the following standard form of a MIP-problem which we will use from now on.

Definition 7. *We say a problem of Mixed Integer Linear Programming has* Relaxed Normalized (RN-) form *if it is of standard form and:*

- *has no continuous variables ($\mathcal{C} = \emptyset$),*
- *lower bound $l_j = 0$ for all variables x_j,*
- *rows $a^T x \leq b$ which fulfill $\|(a^T)\|_\infty = 1$.*

Note that the transformation $P \overset{\text{RN}}{\curvearrowright} \bar{P}$ into RN-form yields an equivalent problem $\bar{P}$ if P is an IP. For MIP, relaxing potentially changes the feasible region of P.

SaP creates a series of RN-form problems. Each of them is obtained by fixing a variable j to a solution value $x_j \leftarrow z_j$. The choice of z_j is subject to the next section.

4.2.2 Best shift selection

After the introduction of the RN-form of a MIP, we need to formulate a criterion to determine when and how to set the solution value of a variable. This section introduces the *best shift* which measures the weighted potential of a variable to violate or satisfy problem rows. Rows are treated independently from each other, a function on the domain of the variable is assigned to every row depending on its feasibility w. r. t. the zero-solution and the potential of the variable to obtain feasibility or cause violations.

Definition 8. *Let P be a MIP in RN-form and $j \in \mathcal{I}$ a variable with domain D_j. For a row $i : a_i^T x \leq b_i$ of P, $i \in \{1, \ldots, m\}$ we define*

$$\Psi_i : D_j \to \{-1, 0, 1\} \tag{4.13}$$

$$t \mapsto \begin{cases} 1, & b_i \geq 0, b_i - a_{ij} \cdot t < 0 \\ -1, & b_i < 0, b_i - a_{ij} \cdot t \geq 0 \\ 0, & \textit{else} \end{cases} \tag{4.14}$$

the row violation function *of row i. The row violation functions of all rows of P sum up to*

$$\Psi : D_j \to \mathbb{Z} \tag{4.15}$$

$$t \mapsto \sum_{i=1}^{m} \Psi_i(t) \tag{4.16}$$

the row violation sum function *of j. For a particular $t \in D_j$ we call $\Psi(t)$ the* violation balance *of t.*

The row violation function Ψ_i of a row i is different from zero for those values of the domain of a variable j, which change the feasibility of i. The definition of the functions implies that rows with coefficient $a_{ij} = 0$ have an assigned row violation function $\Psi_i = 0$.

Example 6 visualizes the idea of the row violation functions and the corresponding row violation sum function.

Example 6. *Let P be a MIP in RN-form with unfixed variable x_0. x_0 appears in the following three rows of the problem:*

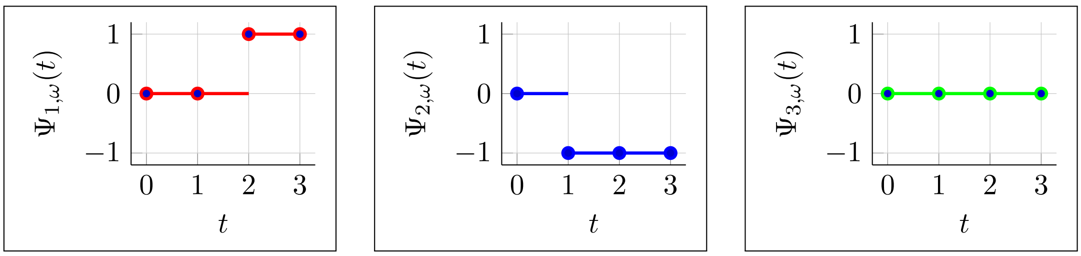
$$\frac{3}{4}x_0 + \ldots \leq 1 \tag{4.17}$$

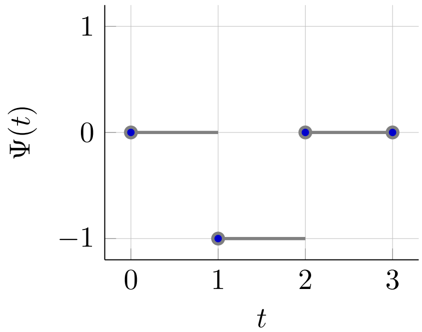
$$-1x_0 + \ldots \leq -\frac{1}{2} \tag{4.18}$$

$$1x_0 + \ldots \leq -1 \tag{4.19}$$

$$x_0 \in \{0, \ldots, 3\} \tag{4.20}$$

The zero solution z_0 is feasible for row (4.17). The respective row violation sum functions Ψ_i shown below reflect how a specific shift of variable x_0 will alter the feasibility of the involved rows.





The resulting row violation sum function $\Psi(t)$ has a minimum $\Psi^* = -1$ for $t^* = 1$. By a shift $\delta x_0 = 1$ the number of infeasible rows is decreased by 1.

The row violation sum function of a variable j evaluates the negative of the number of rows which can be made feasible by setting $x_j \leftarrow t$ for every $0 \le t \le h_j$, and adds the number of rows which would be violated. If $\Psi(t) < 0$, i.e., t has a negative violation balance, setting $x_j \leftarrow t$ decreases the number of infeasible rows of P. Please note that setting $x_j \leftarrow t$ is equivalent to shifting x_j by $\delta x_j = t$ because we assume all unfixed variables implicitly to be set to their lower bound $l_j = 0$. From the previous observations, the minimum of the row violation sum function over the domain of j is of special interest.

Definition 9. *Let P be a MIP in RN-form and $j \in \mathcal{I}$ a variable with domain D_j and row violation sum function Ψ. We call an integer $\delta \in D_j$ with*

$$\Psi(\delta) = \min_{t \in D_j}\{\Psi(t)\} \tag{4.21}$$

best shift *for j.*

We call δ *best shift* because we achieve as many feasible rows as possible by shifting x_j by δ.

Lemma 6 ensures that the row violation sum function has a minimum $\Psi^* = \Psi(\delta x_j)$ for which δx_j will be considered the *best shift* of a variable, even in case of an infinite domain D_j.

Lemma 6. *Let P be a problem in RN-form, $j \in \mathcal{I}$ a variable with domain D_j, and Ψ the row violation sum function of j. Ψ satisfies the following properties:*
a. Ψ has a minimum $\Psi^* := \min_{t \in D_j}\{\Psi(t)\}$.
b. For a row $a_i^T x \le b_i$ with $a_{ij} \neq 0$, we define its feasibility root

$$t_i^* := \begin{cases} \lceil \frac{b_i}{a_j} \rceil, & b_i < 0,\ a_{ij} < 0,\ \lceil \frac{b_i}{a_{ij}} \rceil \le h_j \\ 0, & \textit{else.} \end{cases} \tag{4.22}$$

The feasibility root of a row is the minimum shift value necessary to make the row feasible, or 0, else. With this definition, it holds that

$$\min_{i=1}^{m} \Psi(t_i^*) = \Psi^*. \tag{4.23}$$

Proof. a. We observe that for each row $1 \le i \le m$ the row violation function Ψ_i changes its value at most once within the domain D_j. This can only occur for rows in which j appears with a nonzero coefficient $a_{ij} \neq 0$. This implies that the sum Ψ changes its value at most $|A_j|$ times and one of those finitely many values is necessarily a minimum.

b. If all $t_i^* > 0$, j appears in all of its rows with a negative coefficient $a_{ij} < 0$. Furthermore, all rows in which j appears are currently infeasible because $b_i < 0$. A shift of j by $\delta x_j := \max\{t_i^*\}$ makes all those rows feasible in which j has a nonzero coefficient:

$$b_i - a_{ij} \cdot \delta x_j \ge b_i - a_{ij} \cdot t_i^* \tag{4.24}$$

$$= b_i - a_{ij} \lceil \frac{b_i}{a_{ij}} \rceil \tag{4.25}$$

$$\ge b_i - b_i - a_{ij} \ge 0. \tag{4.26}$$

Making row i feasible adds -1 to $\Psi(t)$ for all $t \geq t_i^*$. Thus, it holds

$$\Psi(\delta x_j) = \sum_{i \in \{1,\ldots,m\}, a_{ij} \neq 0} -1 = -|A_j| \leq \Psi^*. \tag{4.27}$$

because $-|A_j|$ is a lower bound for Ψ^*. This implies

$$\Psi(\delta x_j) = \Psi^*. \tag{4.28}$$

If, however, there exists a row index $\hat{i}$ such that $t_{\hat{i}}^* = 0$, we have

$$\Psi(t_{\hat{i}}^*) = 0. \tag{4.29}$$

In case that $\Psi^* \geq 0$, we are done. If $\Psi^* < 0$ there must be a $t \in D_j$ for which the violation balance $\Psi(t) < 0$ is less than zero. This can only occur if a subset of rows in in which j appears,

$$R := \{i \,|\, b_i < 0,\, b_i - a_{ij}t \geq 0\} \subseteq \{1, \ldots, m\} \tag{4.30}$$

is currently infeasible, but can be made feasible by shifting x_j by a value of t. Then they are already made feasible by shifting x_j by

$$t_{\max}^* := \max\{t_i^* | t_i^* \leq t\} \tag{4.31}$$

and thus

$$\Psi(t_{\max}^*) = \Psi(t) = \Psi^*. \tag{4.32}$$

This proves b.

□

Lemma 6 ensures that Algorithm 6 returns a best shift for a variable j. The running time of Algorithm 6 is

$$\mathcal{O}(|A_j| \log |A_j|). \tag{4.33}$$

Algorithm 6: `bestShift`$(\bar{P}, j, D_j, A_j)$

Input : MIP P in RN-form, integer variable $j \in \mathcal{I}$ with domain D_j and column A_j

Output : Best shift δx_j for j

$Q \leftarrow \emptyset$;
foreach *row* i *with* $a_{ij} \neq 0$ **do**
    if $b_i < 0$,$a_{ij} < 0$ **then**
        $t \leftarrow \lceil \frac{b_i}{a_{ij}} \rceil$;
        if $t \in D_j$ **then** $Q \leftarrow Q \cup (t, -1)$;
    else if $b_i \geq 0$,$a_{ij} > 0$ **then**
        $t \leftarrow \lceil \frac{b_i}{a_{ij}} \rceil$;
        if $t \in D_j$ **then** $Q \leftarrow Q \cup (t, 1)$;
    end
end
if $Q = \emptyset$ **then return** $\delta x_j = 0$;
$\sigma \leftarrow 0$, $\delta x_j \leftarrow 0$, $t_{\text{before}} \leftarrow 0$,s $\Psi^* \leftarrow 0$;
foreach $(t_i, \Psi_i(t_i)) \in Q$ *in nondecreasing order of* t_i **do**
    if $t_i = t_{before}$ **then** $\sigma \leftarrow \sigma + \Psi_i(t_i)$;
    else
        if $\sigma < \Psi^*$ **then**
            $\Psi^* \leftarrow \sigma$, $\delta x_j \leftarrow t_{\text{before}}$;
        $t_{\text{before}} \leftarrow t_i$;
        $\sigma \leftarrow \sigma + \Psi_i(t_i)$;
    end
    $Q \leftarrow Q \setminus \{(t_i, \Psi_i(t_i)\}$;
    if $Q = \emptyset$ *and* $\sigma < \Psi^*$ **then**
        $\Psi^* \leftarrow \sigma$, $\delta x_j \leftarrow t_{\text{before}}$;
    end
end
return δx_j ;

4.2.3 Shift and Propagate

The best shift of a variable is a greedy criterion to choose a promising solution value from the domain of a variable. In this section we specify the order in which we try to shift the variables so as to achieve feasibility. After the choice of an order, we are able to formulate the main heuristic algorithm.

Definition 10. *Let $j \neq k$ be variables of a MIP P in RN-form with matrix A. By*

$$\vartheta_j := \|A_j\|_1 + |A_j| = \sum_{i=1}^{m} |a_{ij}| + |\{i | a_{ij} \neq 0\}|,$$

we define the importance *of j. j is called* more important than k*, if*

$$\vartheta_j > \vartheta_k.$$

The importance is a combined measure of the absolute sum of the coefficients of a variable and its number of nonzeroes. With the importance, we try to measure the overall impact that a variable has on a problem. A variable j has a greater importance for P than k, if it appears in more rows with a nonzero coefficient and if its coefficient absolutes are greater than those of k. Note that the rows are all normalized to have coefficients in $[-1, 1]$ due to the RN-form. This is supposed to make the row coefficients reasonably comparable.

A comparison of different variable sortations is subject to Section 4.3.

The ideas which we have collected are summarized by Algorithm 7.

The SaP heuristic transforms a given MIP P into its IP-relaxation $\bar{P}$ in RN-form with integer variables $\bar{\mathcal{I}}$. The zero solution z_0 is considered as reference solution for which the feasibility for $\bar{P}$ is derived from the right hand side $\bar{b}$ in line 7. If a row i with $\bar{b}_i < 0$ exists, z_0 is not feasible for $\bar{P}$. The next unfixed variable $j \in \bar{\mathcal{I}}$ is selected. The algorithm calculates the best shift δx_j in line 10. The fixation of j, $\bar{\bar{D}}_j \leftarrow \{\delta x_j\}$ expands a new node in the search tree in line 11. If the fixation of j leads to an empty domain $\bar{\bar{D}}_k = \emptyset$, a 1-level backtrack is applied. The variable j remains unfixed in $\bar{P}$, and the next variable is processed.

If, however, the propagation does not lead to an empty domain, we obtain an IP $\bar{\bar{P}}$ in RN-form. The variable j is eliminated from $\bar{\bar{P}}$ by setting and a shrinked variable set $\bar{\bar{\mathcal{I}}} \subseteq \bar{\mathcal{I}} \setminus \{j\}$. A sequence of RN-form problems is generated for which the feasibility of the respective zero solution is checked. In case of a feasible zero solution, the algorithm retransforms all preceding fixations and obtains a solution for the original IP-relaxation of P. If P was an IP, the heuristic terminates with a feasible solution. If, however, P contained continuous variables, a final LP has to be solved after having fixed all integer variables of P to their respective solution values.

The algorithm terminates in line 1, if a free integer variable is detected. The rationale is that we want to avoid overhead in the data structure caused by the introduction of more variables. On our test set, there is no instance with a free integer variable.

Furthermore, the heuristic stops when it exceeds a predefined number of backtracked propagations. The maximum number of backtracking steps when a node was cut off in line 13 is limited by the parameter `cutoffbreaker`. If the heuristic exceeds `cutoffbreaker` backtracking steps, it terminates. If `cutoffbreaker` is set to `-1`, no limitation is given. The number of *propagation rounds* after every fixation is limited by the parameter `nproprounds`. Both parameters are supposed to reduce the overall running time of the heuristic.

We slightly altered the selection of the best shift: if for the row violation sum function Ψ of the processed variable j it holds that $\Psi \equiv 0$, Algorithm 6 returns 0 in line 11. In the final implementation of SaP, the procedure returns either h_j in case of a finite domain D_j or 1 if such a shift increases the slack of all involved feasible rows. It returns 0, if a different shifting value will decrease the slack of some of the involved rows.

Algorithm 7: Shift and Propagate

Input : A MIP P with constraint matrix A and right hand side b
Output : an IP-feasible solution z for P , or *NULL*

Transform $P \curvearrowright \bar{P}$;
foreach *variable* $j \in \bar{\mathcal{I}}$ *of* $\bar{P}$ *in nondecreasing order of importance* **do**
  if $\bar{l}_j = \bar{h}_j$ **then**
    $\bar{x}_j \leftarrow \bar{l}_j$;
    continue;
  end
  if $\bar{b} \geq 0$ **then**
    Stop ;
  else
    $\delta x_j \leftarrow$ `bestShift(`$\bar{P}, j, \bar{D}_j, \bar{A}_j$`)` ;
    $\bar{\bar{P}} \leftarrow \bar{P}$;
    propagate $\bar{\bar{D}}_j \leftarrow \{\delta x_j\}$;
    if $\exists k \in \bar{\bar{\mathcal{I}}} : \bar{\bar{D}}_k = \emptyset$ **then** **backtrack**;
    else
      $\bar{\bar{x}}_j \leftarrow \delta x_j$;
      $\bar{P} \leftarrow \bar{\bar{P}}$;
    end
  end
end
if $\bar{b} \geq 0$ **then**
  foreach $j \in \mathcal{I}$ **do**
    $z_j \leftarrow$ `retransform(`$\bar{l}_j$`)` ;
  end
  if $\mathcal{C} \neq \emptyset$ **then**
    solve the resulting LP ;
    foreach $j \in \mathcal{C}$ **do**
      $z_j \leftarrow z_j^*$;
    end
  end
  if z *IP-feasible for* P **then** **return** z ;
end
return NULL;

Setting	sols	$\varnothing_t$	$sols_{\text{SaP}}$	$\varnothing_{t_{SaP}}$
SaP_Down	116	1.36	72	1.19
SaP_Rand	117	1.44	74	1.16
SaP_Up	121	1.39	77	1.23
SaP_None	120	1.48	71	1.12
SCIP_Default	111	1.56	0	1.00

Table 4.1: Comparison between Shift and Propagate sortation settings and SCIP_Default

4.3 Computational results

Shift and Propagate is a heuristic to find MIP solutions at an early stage of the solution process. We investigate the influence of the SaP heuristic at the root node. Therefore, we set the SCIP `node limit` to `1`. For the tests, the *maximum probing depth* of the heuristic probing tree is set to 65000 which is close to an internal bound of SCIP. Both the parameters `cutoffbreaker` and `nproprounds` are set to `-1` in order to have no limitation of the maximum number of cutoffs during the probing and the number of propagation rounds, respectively. SaP was run with different sortation keys for the order in which variables are processed in Algorithm 7. Namely these are no sortation (SaP_None), a random permutation of the variables (SaP_Rand), and sortations w. r. t. the importance ϑ (see Definition 10) in nonincreasing (SaP_Down) and nondecreasing (SaP_Up) order. The different variants are compared to the SCIP default settings SCIP_Default. Note that SaP is activated by default since the SCIP release version 2.0, hence the default setting has been modified by deactivating the SaP heuristic in contrast to the tests in other chapters of this thesis.

A comparison between the different variants of SaP and SCIP_Default is presented in Table 4.1. The different settings lead to similar results on the test set. The sols column of the table shows that all SaP settings found feasible solutions on up to 9 % more of the test problems than the SCIP_Default setting which was successful on 111 problems. The leading setting in this respect is SaP_Up which finished the root node solving with feasible solutions to 121 problems, followed by SaP_None with 120 problems. The geometric mean time in seconds until the first solution was found is given in column $\varnothing_t$ for the 111 problems for which all solvers found a feasible solution at root node. SCIP_Default took a geometric mean time of 1.56 sec The SaP_Down setting as leading heuristic setting in this respect took 1.36 sec or 13 % less than SCIP_Default. The use of SaP_Up yielded an increase of 11 % over the SCIP_Default.

The heuristic finds solutions on at most 77 instances with SaP_Up. This is 47 % of the test set. With the other settings the heuristic finds solutions on at least 71 or 43.6 % of the test set problems. The geometric mean time spent on the heuristic execution varies between 1.12 sec on average using SaP_None to 1.23 sec using SaP_Up. Recall that all measured times below one second are rounded up to 1.0 sec for the measurement of the geometric mean.

We focus on the setting SaP_Up because this setting found the maximum number of solutions at the root node compared with all other settings.

The heuristic finds a solution on eight instances for which SCIP does not find any solution during the root node, namely `ds`, `flugpl`, `lrn`, `momentum3`, `neos818918`,

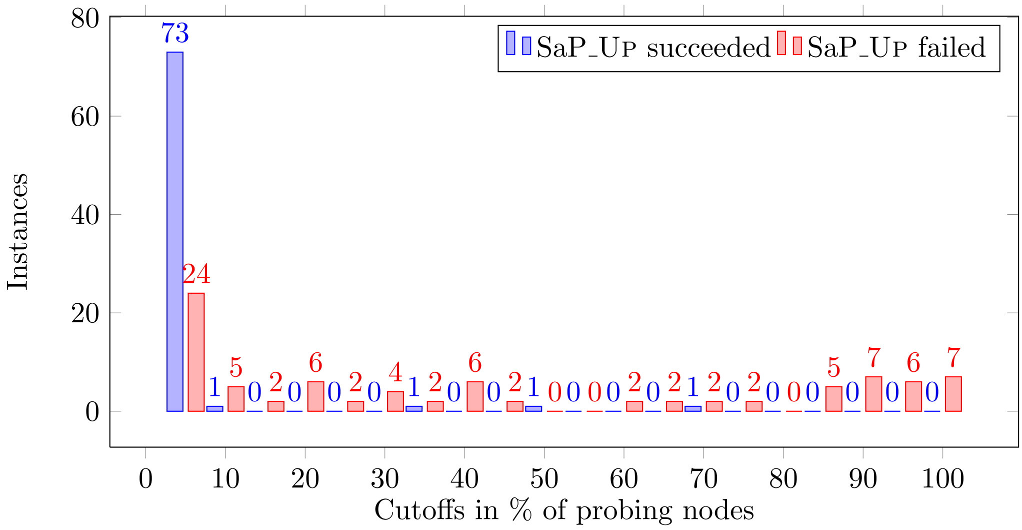


Figure 4.1: Distribution of cutoffs in % of probing nodes for SaP_Up

Setting	$\varnothing_T$	$\varnothing_{\text{nodes}}$	time
SCIP_Default	14.33	391.0	31
SaP_Up	13.75	354.7	30

Table 4.2: The impact of SaP_Up concerning the overall solution process. 125 instances are compared. 34 instances are excluded for time limit reasons.

`ns1648184`, `p0201`, and `swath`. On two more instances, `glass4` and `neos897005`, a solution is found by the SaP_Up setting but not by the Shift and Propagate heuristic itself. On these two instances, the *conflict clauses* produced by the propagation deliver crucial information for other heuristics to find feasible solutions. There is no instance in our test set for which SCIP_Default finds a solution but SaP_Up does not. The running time of the heuristic exceeds one second on 8 of the 163 instances.

Figure 4.1 shows the distribution of instances for which Shift and Propagate with setting SaP_Up succeeded or failed, together with the percentage of cutoffs w. r. t. the total number of created probing nodes. The majority of 73 problems (out of 77 instances where Shift and Propagate suceeded) were solved with a cutoff percentage less than 5 %. Other settings show a very similar distribution.

4.3.1 Summary

The developed algorithm with the SaP_Up setting has contributed feasible solutions to 47 % of our test set. In particular, it has enlarged the set of problems for which SCIP can provide a feasible solution at the root node by 9 %. Furthermore, the use of Shift and Propagate yielded a reduction of the required solving time until a feasible solution is at hand by 11 % compared to SCIP_Default. On two problems, the use of SaP_Up led to feasible solutions found by other heuristics but Shift and Propagate as a side effect of the domain propagation routines and resulting conflict clauses.

Table 4.2 shows the impact of SaP_Up on the overall solving process of SCIP w. r. t. the geometric mean time and the geometric mean number of nodes for 125 instances for which both settings finished within the time limit of 30 min. There is a reduction of 4.1 % regarding the geometric mean time and a reduction of 9.3 % for the number of solving nodes. The column *time* shows the respective number of instances for which the settings hit the time limit.

We have also seen that the occurence of more than 5 % cutoffs is in many cases an indicator that the heuristic will not succeed. Thus we decided to delimit the number of cutoffs by a constant which is currently set to `15` with no dependency on the number of variables of the problem.

Chapter 5

Summary

This thesis introduces three different primal heuristics for MIP. In Chapter 2, we discuss the implementation of the ZI round heuristic. Originally presented in [Wal09], ZI round is an extension of the *simple rounding* heuristic of SCIP (Section 1.4.1). Based on the *fractionality* of a variable ZI round applies a more sophisticated rounding than simple rounding. It outperforms simple rounding in a root node experiment in which it finds more feasible solutions with an objective value smaller or equal to the objective value of a simple rounding solution.

We introduce the 2-opt heuristic in Chapter 3 as improvement heuristic which alters two solution values of variables at the same time in order to improve the objective value. Moreover, we discuss cases of MIPs for which pairwise shifts are desirable. The general Algorithm 4 is generic because it does not specify the variables to be paired to each other by the heuristic. In a practical experiment, we see that the combination of all possible variables to each other can have a deteriorating effect on the overall solving performance.

We introduce a more restrictive neighborhood which measures the ratio of shared rows between two variables. Furthermore, we present a fast algorithm to calculate the desired neighborhoods. The introduction of the neighborhoods yields a dramatic reduction of the time required by 2-opt, and they turn out to be flexible because they contain the combination of all possible pairs as a special case.

Finally, Chapter 4 deals with the sophisticated *propagation heuristic* Shift and Propagate, which can be called before the root LP has been solved. The heuristic expands an auxiliary branch-and-bound tree in which it fixes one variable per node. It then propagates this domain reduction to deduce further domain reductions or to detect infeasible sub-problems. We see that both cases can be beneficial. We observe that the heuristic contributes feasible solutions to up to 47 % of our benchmark instances and can reduce the average solving time by 5 % and the average number of solving nodes by even almost 10 %.

The implementations have been done with the highest possible accuracy. Yet, they represent their current implementations in SCIP, rather than final algorithms with no space left for improvement. The complexity of MIP favors the design of new heuristics or the refinement of existing ones so that primal heuristic remain a topic for future research.

Chapter 6

Appendix

	SCIP 2.0.1					SCIP 2.0.1 no heuristics				
Problem Name	T	t_0	t_{opt}	nodes	T-limit	T	t_0	t_{opt}	nodes	T-limit
10teams	8.76	3.89	8.76	844		9.30	6.37	9.30	697	
30:70:4_5:0_5:100	135.57	0.04	135.56	34		239.37	239.36	239.36	924	
30:70:4_5:0_95:98	97.69	0.04	97.68	142		70.97	70.96	70.96	138	
30:70:4_5:0_95:100	126.67	0.05	126.66	16		143.50	143.49	143.49	342	
a1c1s1	1 800.00	0.21	243.09	113 381	time	1 800.00	571.47	571.47	144 433	time
acc-0	20.67	20.67	20.67	121		15.22	15.22	15.22	187	
acc-1	1.09	1.09	1.09	1		33.72	33.72	33.72	100	
acc-2	69.39	69.38	69.38	55		55.64	55.64	55.64	100	
acc-3	173.24	173.23	173.23	177		109.82	109.82	109.82	131	
acc-4	116.36	116.36	116.36	48		790.24	790.23	790.23	1 441	
acc-5	504.81	504.81	504.81	1 071		604.64	604.64	604.64	2 582	
acc-6	51.34	51.34	51.34	65		266.17	266.17	266.17	1 167	
aflow30a	12.00	0.09	8.19	2 993		17.95	6.20	16.05	4 862	
aflow40b	1 504.23	0.83	446.24	285 246		1 800.00	34.45	650.75	243 832	time
air03	29.89	28.49	29.89	1		28.81	28.81	28.81	1	
air04	47.78	38.07	47.12	368		39.75	35.64	39.63	136	
air05	23.88	6.79	22.67	172		24.70	20.41	23.84	306	
arki001	1 450.67	2.43	1 439.65	929 641		1 800.00	285.19	1 708.98	1 031 127	time
atlanta-ip	1 800.00	600.58	1 703.98	2 118	time	1 800.00	768.85	1 669.39	2 259	time
bc1	210.30	5.90	175.45	6 026		240.01	7.73	213.98	7 801	
bell3a	10.17	0.02	0.03	44 709		11.01	0.04	1.03	49 233	
bell5	0.35	0.01	0.12	1 317		0.29	0.09	0.12	1 722	
bienst1	25.39	0.67	13.14	21 091		17.63	2.80	15.52	15 226	
bienst2	179.83	1.17	100.62	120 858		100.62	3.22	24.84	89 778	
binkar10_1	200.90	1.47	148.06	129 894		180.45	3.34	103.27	136 853	
blend2	0.45	0.24	0.24	163		1.90	1.18	1.75	4 737	
cap6000	2.60	0.30	2.27	2 397		1 800.00	530.26	543.41	625 902	time
dano3_3	124.14	7.76	124.14	7		91.30	91.19	91.19	43	
dano3_4	154.97	5.62	139.00	10		115.74	115.30	115.30	115	
dano3_5	315.90	12.06	303.55	196		290.50	265.28	270.84	773	
dano3mip	1 800.00	2.31	1 516.24	1 259	time	1 800.02			2 231	time
danoint	1 800.00	0.03	29.28	243 817	time	1 800.00	10.99	111.50	384 848	time
dcmulti	0.90	0.03	0.87	7		1.21	0.93	1.20	82	

continued on next page

Problem Name	SCIP 2.0.1 T	t_0	t_{opt}	nodes	T-limit	SCIP 2.0.1 no heuristics T	t_0	t_{opt}	nodes	T-limit
disctom	2.96	2.96	2.96	1		1 800.00			117 202	time
ds	1 800.06	52.32	1 227.24	150	time	1 800.02	1 682.16	1 682.16	849	time
dsbmip	0.15	0.15	0.15	1		0.30	0.29	0.30	6	
egout	0.02	0.01	0.02	1		0.02	0.02	0.02	1	
eilD76	18.31	1.85	18.31	3		17.08	14.98	17.08	3	
enigma	0.07	0.07	0.07	75		0.21	0.19	0.21	665	
fast0507	270.48	0.30	129.08	997		275.06	136.28	177.57	1 367	
fiber	0.94	0.03	0.61	11		2.22	0.85	2.21	272	
fixnet6	1.79	0.04	1.78	15		1.62	1.46	1.60	78	
flugpl	0.02	0.02	0.02	76		0.02	0.02	0.02	107	
gen	0.08	0.03	0.08	1		0.06	0.06	0.06	1	
gesa2-o	1.01	0.14	1.01	1		1.53	1.40	1.50	136	
gesa2	1.17	0.06	1.17	11		1.49	1.33	1.47	108	
gesa3	1.13	0.05	1.13	17		2.32	2.18	2.25	199	
gesa3_o	1.63	1.28	1.57	11		2.94	2.75	2.93	127	
glass4	1 219.34	0.65	1 007.49	1 955 670		966.71	1.24	468.95	1 889 673	
gt2	0.08	0.02	0.07	1		0.07	0.07	0.07	18	
harp2	141.99	0.80	86.84	161 634		557.12	5.28	457.74	685 799	
irp	48.09	3.57	47.38	451		29.64	10.19	20.70	275	
khb05250	0.45	0.02	0.37	1		0.54	0.53	0.53	9	
l152lav	2.31	0.92	2.19	54		2.26	0.80	2.16	67	
liu	1 800.00	0.06	824.64	704 414	time	1 495.38			1 180 460	
lseu	0.22	0.01	0.21	384		0.12	0.06	0.12	166	
manna81	0.41	0.02	0.41	1		0.33	0.33	0.33	1	
markshare1	1 800.00	0.01	47.41	17 017 924	time	1 800.00	0.04	1 636.57	18 847 238	time
markshare2	1 239.87	0.01	329.13	10 237 514		1 605.41	0.07	499.20	14 801 378	
markshare2_1	1 800.00	0.01	703.61	20 773 334	time	691.84	0.03	691.30	9 418 832	
markshare4_0	151.91	0.00	63.77	1 978 280		102.82	0.01	7.95	1 513 842	
mas74	624.82	0.01	108.36	3 368 786		528.97	0.28	107.21	3 197 323	
mas76	56.21	0.01	0.53	330 335		50.88	0.40	22.20	306 595	
mas284	10.56	0.06	6.11	15 979		9.51	1.49	6.84	19 023	
mik.250-20-75.1	3.31	0.01	0.71	9 089		13.16	1.20	4.19	41 286	
mik.250-20-75.2	2.45	0.00	0.77	5 031		7.89	1.34	2.69	23 082	
mik.250-20-75.3	2.51	0.01	0.84	4 597		11.54	1.33	2.87	35 957	

continued on next page

	SCIP 2.0.1					SCIP 2.0.1 no heuristics				
Problem Name	T	t_0	t_{opt}	nodes	T-limit	T	t_0	t_{opt}	nodes	T-limit
mik.250-20-75.4	27.37	0.00	0.82	102 249		36.59	1.76	5.73	115 827	
mik.250-20-75.5	4.64	0.01	0.99	13 349		18.76	1.59	6.49	56 952	
misc03	0.98	0.37	0.52	67		0.92	0.31	0.59	107	
misc06	0.26	0.05	0.26	5		0.20	0.19	0.20	27	
misc07	16.67	0.24	0.64	26 110		20.05	0.27	0.42	34 924	
mitre	6.06	5.72	6.06	1		6.22	6.22	6.22	1	
mkc	1 800.00	0.01	831.71	682 229	time	757.69	22.07	339.27	480 616	
mkc1	1 800.00	0.01	5.91	634 169	time	1 800.00	6.50	21.36	714 231	time
mod008	0.65	0.00	0.03	343		1.17	0.50	1.17	1 458	
mod010	1.05	0.76	1.05	1		1.01	0.86	0.98	18	
mod011	63.82	0.03	52.48	1 899		71.29	15.01	65.90	2 450	
modglob	0.60	0.01	0.60	84		0.67	0.63	0.65	247	
momentum1	1 800.01			2 226	time	1 800.01			3 109	time
momentum2	1 800.01	518.34	1 101.51	4 125	time	1 800.01			5 673	time
momentum3	1 800.00			1	time	1 801.46			1	time
msc98-ip	1 800.00	1 569.82	1 569.82	24	time	1 800.01			416	time
mzzv11	282.69	0.04	264.15	2 559		303.19	118.80	174.10	3 557	
mzzv42z	146.06	0.06	143.38	714		154.56	110.50	152.67	1 160	
neos1	1.58	0.10	0.20	1		2.26	2.26	2.26	1	
neos2	45.81	7.40	45.49	16 448		79.75	6.58	78.95	35 564	
neos3	1 619.21	7.58	1 592.47	372 284		1 800.00	12.01	1 446.19	397 014	time
neos4	2.53	2.50	2.53	1		2.46	2.46	2.46	1	
neos5	1 340.89	0.00	167.46	6 949 722		911.70	0.16	11.18	4 977 491	
neos6	111.96	45.22	111.96	2 069		748.95	50.67	748.94	45 661	
neos7	65.42	0.24	1.80	45 256		115.21	1.84	40.26	36 389	
neos8	14.44	0.15	14.44	1		14.31	14.31	14.31	1	
neos9	1 800.01	310.43	396.98	1 882	time	610.89	101.77	155.10	3 732	
neos10	20.69	16.21	20.69	9		20.44	19.53	20.44	9	
neos11	226.29	14.42	22.75	2 703		369.10	21.13	40.09	6 789	
neos12	1 105.41	70.73	236.78	2 406		530.63	50.51	59.58	2 136	
neos13	185.23	2.22	151.68	1 601		1 800.01	22.76	1 601.83	63 299	time
neos16	1 800.00	276.59	320.78	1 115 504	time	1 800.00	149.29	687.50	1 042 341	time
neos20	6.08	3.84	5.95	601		5.48	4.54	4.74	861	
neos21	27.07	0.03	12.37	2 117		29.90	9.55	14.82	3 429	

continued on next page

Problem Name	SCIP 2.0.1 T	t_0	t_{opt}	nodes	T-limit	SCIP 2.0.1 no heuristics T	t_0	t_{opt}	nodes	T-limit
neos22	0.64	0.19	0.64	1		1.34	1.34	1.34	1	
neos23	8.63	0.08	2.52	8 549		9.31	2.93	3.65	9 774	
neos616206	1 800.00	12.60	89.77	983 231	time	1 800.00	52.56	110.01	1 158 774	time
neos632659	62.40	0.01	22.99	172 401		20.21	0.36	0.36	63 138	
neos648910	1.13	0.06	0.31	38		5.42	5.41	5.42	6 028	
neos808444	1 800.01			153	time	1 094.17	1 094.14	1 094.14	832	
neos818918	1 800.00	0.05	21.21	618 708	time	1 800.00	3.97	192.41	665 850	time
neos823206	585.42	54.27	155.38	22 488		629.77	56.03	178.89	30 347	
neos897005	33.64	33.61	33.61	1		204.25	204.23	204.23	31	
net12	1 800.00	179.98	994.94	3 426	time	1 800.00	156.90	192.20	3 820	time
noswot	229.93	0.01	77.99	611 086		230.91	0.12	0.88	690 191	
ns1648184	1 800.00	2.68	1 111.50	302 191	time	1 800.00	11.40	217.53	254 109	time
ns1688347	456.52	90.60	456.52	8 335		1 800.00	32.43	121.77	24 448	time
ns1671066	1 800.00	0.21	614.07	687 907	time	1 800.00	6.38	828.89	887 496	time
ns1692855	1 800.00	176.09	454.55	10 754	time	1 800.00	168.13	168.13	20 458	time
nug08	130.14	1.78	130.14	1		130.21	130.21	130.21	1	
nw04	93.75	49.42	89.98	5		76.62	70.09	71.19	5	
opt1217	0.33	0.01	0.03	1		1 800.00	2.93	3.19	3 007 717	time
p0033	0.02	0.01	0.02	1		0.02	0.02	0.02	1	
p0201	0.80	0.41	0.57	68		0.94	0.35	0.69	216	
p0282	0.43	0.00	0.43	10		0.90	0.79	0.85	302	
p0548	0.22	0.06	0.21	68		0.34	0.32	0.34	87	
p2756	1.73	0.18	1.73	187		1.93	1.46	1.93	151	
pk1	55.55	0.01	15.76	247 509		57.90	0.23	21.75	323 765	
pp08a	1.04	0.01	0.89	573		0.87	0.74	0.77	743	
pp08aCUTS	0.97	0.02	0.51	73		0.92	0.72	0.83	453	
prod1	14.84	0.04	2.04	24 615		11.63	4.62	4.62	23 433	
prod2	43.69	0.13	41.73	49 024		41.76	40.89	41.48	76 799	
protfold	1 800.01	647.67	647.67	3 922	time	1 800.00			2 246	time
qap10	213.64	8.84	184.63	1		214.91	197.37	197.37	3	
qiu	59.12	0.05	6.32	12 750		66.72	7.51	8.33	13 955	
qnet1	1.76	0.13	1.75	13		2.95	2.64	2.91	134	
qnet1_o	1.70	0.04	1.70	28		1.98	0.99	1.98	63	
ran8x32	10.47	0.01	4.40	11 965		11.05	4.33	4.33	20 006	

continued on next page

Problem Name	SCIP 2.0.1 T	t_0	t_{opt}	nodes	T-limit	SCIP 2.0.1 no heuristics T	t_0	t_{opt}	nodes	T-limit
ran10x26	31.49	0.01	17.24	39 082		25.69	4.51	15.02	43 172	
ran12x21	64.68	0.02	11.97	86 000		60.99	4.04	8.79	94 649	
ran13x13	32.19	0.01	19.70	43 317		29.03	3.07	9.59	53 012	
rd-rplusc-21	1 800.00	227.31	1 343.79	40 146	time	1 800.01	171.70	701.85	47 819	time
rentacar	1.68	1.65	1.67	9		2.50	2.48	2.49	13	
rgn	0.23	0.01	0.17	1		0.24	0.16	0.16	114	
roll3000	1 800.00	3.77	412.77	455 276	time	1 800.00	13.38	513.13	446 296	time
rout	34.89	0.07	19.02	24 218		44.01	2.92	36.98	38 799	
set1ch	0.52	0.03	0.52	21		0.49	0.43	0.48	25	
seymour	1 800.01	0.01	1 784.54	35 137	time	1 800.00	59.82	575.75	46 911	time
seymour1	236.32	0.01	52.10	4 626		215.87	14.05	47.10	4 617	
sp97ar	1 800.01	2.18	1 446.01	9 879	time	1 800.00	203.20	203.20	26 325	time
stein27	0.81	0.00	0.17	3 963		0.60	0.11	0.11	4 433	
stein45	15.80	0.00	1.08	52 789		13.04	0.67	1.09	52 504	
stp3d	1 801.16			2	time	1 805.60			7	time
swath1	26.75	8.65	26.47	741		52.98	11.05	51.07	3 511	
swath2	43.46	11.61	40.89	2 125		54.00	9.97	49.91	3 589	
swath3	194.85	12.16	79.09	18 239		515.46	11.30	334.88	55 210	
t1717	1 800.03	352.65	635.77	623	time	1 800.04			1 568	time
timtab1	635.20	1.11	59.29	941 228		463.07	14.83	27.54	738 188	
timtab2	1 800.00	6.67	1 602.76	1 519 033	time	1 800.00	220.43	557.58	1 639 925	time
tr12-30	1 658.27	0.38	226.12	1 155 662		1 800.00	13.10	341.55	1 436 597	time
vpm1	0.02	0.00	0.02	1		0.04	0.04	0.04	1	
vpm2	0.90	0.01	0.52	399		1.25	0.64	1.00	2 450	

Table 6.1: Results for SCIP 2.0.1 with and without primal heuristics

Problem Name	Primal Bound	Optimum	Δ
10teams	924.0	924.0	0.0000
30:70:4_5:0_5:100	9.0	9.0	0.0000
30:70:4_5:0_95:98	12.0	12.0	0.0000
30:70:4_5:0_95:100	3.0	3.0	0.0000
a1c1s1	11 562.0	11 503.4	0.5093
acc-0	0.0	0.0	0.0000
acc-1	0.0	0.0	0.0000
acc-2	0.0	0.0	0.0000
acc-3	0.0	0.0	0.0000
acc-4	0.0	0.0	0.0000
acc-5	0.0	0.0	0.0000
acc-6	0.0	0.0	0.0000
aflow30a	1 158.0	1 158.0	0.0000
aflow40b	1 168.0	1 168.0	0.0000
air03	$3.4 \cdot 10^5$	$3.4 \cdot 10^5$	0.0000
air04	56 137.0	56 137.0	0.0000
air05	26 374.0	26 374.0	0.0000
arki001	$7.6 \cdot 10^6$	$7.6 \cdot 10^6$	0.0000
atlanta-ip	99.0	90.0	9.9988
bc1	3.3	3.3	0.0000
bell3a	$8.8 \cdot 10^5$	$8.8 \cdot 10^5$	0.0000
bell5	$9.0 \cdot 10^6$	$9.0 \cdot 10^6$	0.0000
bienst1	46.8	46.8	0.0000
bienst2	54.6	54.6	0.0000
binkar10_1	6 742.2	6 742.2	0.0000
blend2	7.6	7.6	0.0000
cap6000	$-2.5 \cdot 10^6$	$-2.5 \cdot 10^6$	0.0000
dano3_3	576.3	576.3	0.0000
dano3_4	576.4	576.4	0.0000
dano3_5	576.9	576.9	0.0000
dano3mip	712.9		∞
danoint	65.7	65.7	0.0000
dcmulti	$1.9 \cdot 10^5$	$1.9 \cdot 10^5$	0.0000
disctom	$-5\,000.0$	$-5\,000.0$	0.0000
ds	404.2		∞
dsbmip	-305.2	-305.2	0.0000
egout	568.1	568.1	0.0000
eilD76	885.4	885.4	0.0000
enigma	0.0	0.0	0.0000
fast0507	174.0	174.0	0.0000
fiber	$4.1 \cdot 10^5$	$4.1 \cdot 10^5$	0.0000
fixnet6	3 983.0	3 983.0	0.0000
flugpl	$1.2 \cdot 10^6$	$1.2 \cdot 10^6$	0.0000
gen	$1.1 \cdot 10^5$	$1.1 \cdot 10^5$	0.0000
gesa2-o	$2.6 \cdot 10^7$	$2.6 \cdot 10^7$	0.0000
gesa2	$2.6 \cdot 10^7$	$2.6 \cdot 10^7$	0.0000
gesa3	$2.8 \cdot 10^7$	$2.8 \cdot 10^7$	0.0000
gesa3_o	$2.8 \cdot 10^7$	$2.8 \cdot 10^7$	0.0000
glass4	$1.6 \cdot 10^9$	$1.2 \cdot 10^9$	33.3331
gt2	21 166.0	21 166.0	0.0000
harp2	$-7.4 \cdot 10^7$	$-7.4 \cdot 10^7$	0.0000
irp	12 159.5	12 159.5	0.0000
khb05250	$1.1 \cdot 10^8$	$1.1 \cdot 10^8$	0.0000
l152lav	4 722.0	4 722.0	0.0000
liu	1 256.0		∞
lseu	1 120.0	1 120.0	0.0000
manna81	$-13\,164.0$	$-13\,164.0$	0.0000

continued on next page

Problem Name	Primal Bound	Optimum	Δ
markshare1	6.0	1.0	500.0000
markshare2	19.0	1.0	1 800.0000
markshare2_1	$1.8 \cdot 10^{-4}$	0.0	∞
markshare4_0	1.0	1.0	0.0000
mas74	11 801.2	11 801.2	0.0000
mas76	40 005.1	40 005.1	0.0000
mas284	91 405.7	91 405.7	0.0000
mik.250-20-75.1	−49 716.0	−49 716.0	0.0000
mik.250-20-75.2	−50 768.0	−50 768.0	0.0000
mik.250-20-75.3	−52 242.0	−52 242.0	0.0000
mik.250-20-75.4	−52 301.0	−52 301.0	0.0000
mik.250-20-75.5	−51 532.0	−51 532.0	0.0000
misc03	3 360.0	3 360.0	0.0000
misc06	12 850.9	12 850.9	0.0000
misc07	2 810.0	2 810.0	0.0000
mitre	$1.2 \cdot 10^{5}$	$1.2 \cdot 10^{5}$	0.0000
mkc	−552.2	−563.8	2.0633
mkc1	−607.2	−607.2	0.0000
mod008	307.0	307.0	0.0000
mod010	6 548.0	6 548.0	0.0000
mod011	$-5.5 \cdot 10^{7}$	$-5.5 \cdot 10^{7}$	0.0000
modglob	$2.1 \cdot 10^{7}$	$2.1 \cdot 10^{7}$	0.0000
momentum1		$1.1 \cdot 10^{5}$	∞
momentum2	15 416.0	12 314.2	25.1889
momentum3			∞
msc98-ip	$2.9 \cdot 10^{7}$	$2.0 \cdot 10^{7}$	44.4810
mzzv11	−21 718.0	−21 718.0	0.0000
mzzv42z	−20 540.0	−20 540.0	0.0000
neos1	19.0	19.0	0.0000
neos2	454.9	454.9	0.0000
neos3	368.8	368.8	0.0000
neos4	$-4.9 \cdot 10^{10}$	$-4.9 \cdot 10^{10}$	0.0000
neos5	15.0	15.0	0.0000
neos6	83.0	83.0	0.0000
neos7	$7.2 \cdot 10^{5}$	$7.2 \cdot 10^{5}$	0.0000
neos8	−3 719.0	−3 719.0	0.0000
neos9	798.0	798.0	0.0000
neos10	−1 135.0	−1 135.0	0.0000
neos11	9.0	9.0	0.0000
neos12	13.0	13.0	0.0000
neos13	−95.5	−95.5	0.0000
neos16	446.0		∞
neos20	−434.0	−434.0	0.0000
neos21	7.0	7.0	0.0000
neos22	$7.8 \cdot 10^{5}$	$7.8 \cdot 10^{5}$	0.0000
neos23	137.0	137.0	0.0000
neos616206	937.6	937.6	0.0000
neos632659	−94.0	−94.0	0.0000
neos648910	32.0	32.0	0.0000
neos808444		0.0	∞
neos818918	1 700.0	1 700.0	0.0000
neos823206	83.9	83.9	0.0000
neos897005	14.0	14.0	0.0000
net12	214.0	214.0	0.0000
noswot	−41.0	−41.0	0.0000
ns1648184	−1 231.2	−1 231.3	0.0110
ns1688347	27.0	27.0	0.0000

continued on next page

Problem Name	Primal Bound	Optimum	Δ
ns1671066	7.6	7.6	0.0655
ns1692855	33.0	27.0	22.2222
nug08	214.0	214.0	0.0000
nw04	16 862.0	16 862.0	0.0000
opt1217	−16.0	−16.0	0.0000
p0033	3 089.0	3 089.0	0.0000
p0201	7 615.0	7 615.0	0.0000
p0282	$2.6 \cdot 10^5$	$2.6 \cdot 10^5$	0.0000
p0548	8 691.0	8 691.0	0.0000
p2756	3 124.0	3 124.0	0.0000
pk1	11.0	11.0	0.0000
pp08a	7 350.0	7 350.0	0.0000
pp08aCUTS	7 350.0	7 350.0	0.0000
prod1	−56.0	−56.0	0.0000
prod2	−62.0	−62.0	0.0000
protfold	−21.0	−31.0	32.2581
qap10	340.0	340.0	0.0000
qiu	−132.9	−132.9	0.0000
qnet1	16 029.7	16 029.7	0.0000
qnet1_o	16 029.7	16 029.7	0.0000
ran8x32	5 247.0	5 247.0	0.0000
ran10x26	4 270.0	4 270.0	0.0000
ran12x21	3 664.0	3 664.0	0.0000
ran13x13	3 252.0	3 252.0	0.0000
rd-rplusc-21	$1.7 \cdot 10^5$	$1.7 \cdot 10^5$	1.3201
rentacar	$3.0 \cdot 10^7$	$3.0 \cdot 10^7$	0.0000
rgn	82.2	82.2	0.0000
roll3000	12 920.0	12 890.0	0.2327
rout	1 077.6	1 077.6	0.0000
set1ch	54 537.8	54 537.8	0.0000
seymour	425.0	423.0	0.4728
seymour1	410.8	410.8	0.0000
sp97ar	$6.8 \cdot 10^8$	$6.6 \cdot 10^8$	3.1824
stein27	18.0	18.0	0.0000
stein45	30.0	30.0	0.0000
stp3d			∞
swath1	379.1	379.1	0.0000
swath2	385.2	385.2	0.0000
swath3	397.8	397.8	0.0000
t1717	$1.9 \cdot 10^5$		∞
timtab1	$7.6 \cdot 10^5$	$7.6 \cdot 10^5$	0.0000
timtab2	$1.2 \cdot 10^6$	$1.1 \cdot 10^6$	6.0807
tr12-30	$1.3 \cdot 10^5$	$1.3 \cdot 10^5$	0.0000
vpm1	20.0	20.0	0.0000
vpm2	13.8	13.8	0.0000

Table 6.2: SCIP results on the test set. The column *optimum* is empty, if we do not know the optimal solution of this instance. The gap Δ to the optimal solution is not given in those columns.

Problem Name	Optimal solution	Simplerounding objective	Simplerounding Time	ZI objective	ZI time	ZI loops
30:70:4_5:0_5:100	9.0		0.00	1 276.0	0.01	4
30:70:4_5:0_95:98	12.0		0.00	1 218.0	0.01	4
30:70:4_5:0_95:100	3.0		0.00	1 385.0	0.01	5
bc1	3.3		0.00	4.5	0.00	1
bell3a	$8.8 \cdot 10^5$	$9.2 \cdot 10^5$	0.00	$9.2 \cdot 10^5$	0.00	1
bell5	$9.0 \cdot 10^6$		0.00	$9.2 \cdot 10^6$	0.00	2
cap6000	$-2.5 \cdot 10^6$	$-2.4 \cdot 10^6$	0.00	$-2.4 \cdot 10^6$	0.00	1
egout	568.1	625.3	0.00	625.3	0.00	1
fast0507	174.0	541.0	0.02	304.0	0.02	1
fixnet6	3 983.0	4 536.0	0.00	4 536.0	0.00	1
khb05250	$1.1 \cdot 10^8$	$1.3 \cdot 10^8$	0.00	$1.3 \cdot 10^8$	0.00	1
manna81	−13 164.0	−12 880.0	0.01	−13 157.0	0.00	1
markshare1	1.0	854.0	0.00	541.0	0.00	1
markshare1_1	0.0	803.1	0.00	423.1	0.00	1
markshare2	1.0	1 714.0	0.00	313.0	0.00	1
markshare2_1	0.0	1 714.0	0.00	313.0	0.00	1
markshare4_0	1.0	339.0	0.00	185.0	0.00	1
mik.250-20-75.1	−49 716.0	$1.4 \cdot 10^{-11}$	0.00	$1.8 \cdot 10^{-11}$	0.00	1
mik.250-20-75.2	−50 768.0	$-4.1 \cdot 10^{-12}$	0.00	$2.7 \cdot 10^{-12}$	0.00	1
mik.250-20-75.3	−52 242.0	$-6.1 \cdot 10^{-12}$	0.00	$-5.7 \cdot 10^{-13}$	0.00	1
mik.250-20-75.4	−52 301.0	$-9.7 \cdot 10^{-12}$	0.00	$1.2 \cdot 10^{-11}$	0.00	1
mik.250-20-75.5	−51 532.0	$1.9 \cdot 10^{-11}$	0.00	$3.2 \cdot 10^{-12}$	0.00	1
misc06	12 850.9	12 888.6	0.00	12 888.6	0.00	1
mod008	307.0	510.0	0.00	370.0	0.00	1
mod011	$-5.5 \cdot 10^7$		0.00	$-4.3 \cdot 10^7$	0.00	1
modglob	$2.1 \cdot 10^7$	$2.1 \cdot 10^7$	0.00	$2.1 \cdot 10^7$	0.00	1
neos5	15.0	35.3	0.00	17.3	0.00	1
neos21	7.0		0.00	33.0	0.00	2
nsrand-ipx	51 200.0		0.00	91 680.0	0.01	2
p0282	$2.6 \cdot 10^5$		0.00	$3.2 \cdot 10^5$	0.00	1
pp08a	7 350.0	15 000.0	0.00	15 000.0	0.00	1
pp08aCUTS	7 350.0	16 630.4	0.00	16 630.4	0.01	1
qiu	−132.9	1 805.2	0.00	1 805.2	0.00	1
qnet1_o	16 029.7	28 462.1	0.00	28 462.1	0.00	1

continued on next page

Problem Name	Optimal solution	Simplerounding objective	Simplerounding Time	ZI objective	ZI time	ZI loops
ran8x32	5 247.0	5 837.0	0.00	5 837.0	0.00	1
ran10x26	4 270.0	4 745.0	0.00	4 745.0	0.00	1
ran12x21	3 664.0	4 080.0	0.00	4 080.0	0.00	1
ran13x13	3 252.0	3 521.0	0.00	3 521.0	0.01	1
set1ch	54 537.8	$1.1 \cdot 10^5$	0.00	$1.1 \cdot 10^5$	0.00	1
seymour	423.0	746.0	0.00	590.0	0.00	1
seymour1	410.8	460.8	0.00	443.8	0.00	1
sp97ar	$6.6 \cdot 10^8$		0.00	$9.2 \cdot 10^8$	0.00	11
stein27	18.0	27.0	0.00	19.0	0.00	1
stein45	30.0	45.0	0.00	35.0	0.00	1
vpm1	20.0	23.0	0.00	23.0	0.00	1

Table 6.3: Results of the first comparison experiment between ZI round and Simple Rounding

Problem Name	Time/sec.	Nodes	ZI Time/sec.	Solutions ZI-1	Time limit	ZI1	ZI2	ZI5	Default: Time/sec.	Nodes
10teams	8.71	844	0.00	0		0	0	0	8.67	844
30:70:4_5:0_5:100	135.66	34	0.16	33		0	0	33	133.65	34
30:70:4_5:0_95:98	97.66	142	0.83	138		7	7	138	96.04	142
30:70:4_5:0_95:100	125.19	16	0.11	15		0	0	15	124.44	16
a1c1s1	1 800.00	114 583	0.27	0	time	0	0	0	1 800.00	115 621
acc-0	20.69	121	0.03	0		0	0	0	20.64	121
acc-1	1.08	1	0.00	0		0	0	0	1.08	1
acc-2	69.25	55	0.02	0		0	0	0	68.99	55
acc-3	173.05	177	0.00	0		0	0	0	172.03	177
acc-4	115.96	48	0.00	0		0	0	0	115.53	48
acc-5	504.24	1 071	0.06	0		0	0	0	503.67	1 071
acc-6	51.42	65	0.01	0		0	0	0	51.31	65
aflow30a	11.99	2 993	0.03	0		0	0	0	11.97	2 993
aflow40b	1 507.06	285 246	6.65	0		0	0	0	1 473.07	285 246
air03	28.92	1	0.00	0		0	0	0	28.75	1
air04	47.34	368	0.02	0		0	0	0	46.96	368
air05	23.56	172	0.00	0		0	0	0	23.45	172
arki001	1 440.60	929 641	12.54	0		0	0	0	1 408.09	929 641
atlanta-ip	1 800.01	2 145	1.48	29	time	30	29	29	1 800.01	2 197
bc1	209.91	6 026	0.22	305		305	305	305	203.83	5 989
bell3a	10.29	44 707	0.02	127		127	127	127	10.18	44 709
bell5	0.35	1 317	0.01	232		231	231	232	0.35	1 317
bienst1	25.25	21 091	0.13	0		0	0	0	25.22	21 091
bienst2	178.77	120 858	0.58	0		0	0	0	177.68	120 858
binkar10_1	198.02	129 894	0.99	0		0	0	0	194.53	129 894
blend2	0.45	163	0.00	0		0	0	0	0.45	163
cap6000	2.59	2 397	0.02	192		183	183	192	2.58	2 397
dano3_3	123.50	7	0.01	0		0	0	0	122.99	7
dano3_4	154.64	10	0.00	0		0	0	0	153.50	10
dano3_5	321.65	196	0.00	0		0	0	0	313.17	196
dano3mip	1 800.00	1 249	0.20	0	time	0	0	0	1 800.00	1 264
danoint	1 800.00	244 045	3.19	0	time	0	0	0	1 800.00	245 320
dcmulti	0.91	7	0.00	0		0	0	0	0.90	7
disctom	2.94	1	0.00	0		0	0	0	2.93	1

continued on next page

Problem Name	Time/sec.	Nodes	ZI Time/sec.	Solutions ZI-1	Time limit	ZI1	ZI2	ZI5	Default: Time/sec.	Nodes
ds	1 800.00	151	0.23	0	time	0	0	0	1 800.00	151
dsbmip	0.15	1	0.00	0		0	0	0	0.15	1
egout	0.02	1	0.00	0		0	0	0	0.02	1
eilD76	18.16	3	0.00	0		0	0	0	18.08	3
enigma	0.08	75	0.00	0		0	0	0	0.08	75
fast0507	265.46	1 066	4.68	239		239	239	239	254.74	997
fiber	0.94	11	0.00	1		1	1	1	0.93	11
fixnet6	2.00	15	0.00	8		8	8	8	1.80	15
flugpl	0.02	76	0.00	0		0	0	0	0.03	76
gen	0.07	1	0.00	0		0	0	0	0.07	1
gesa2-o	1.02	1	0.00	0		0	0	0	1.03	1
gesa2	1.18	11	0.00	0		0	0	0	1.17	11
gesa3	1.13	17	0.00	0		0	0	0	1.11	17
gesa3_o	1.64	11	0.00	0		0	0	0	1.62	11
glass4	1 215.90	1 955 670	6.61	0		0	0	0	1 197.22	1 955 670
gt2	0.07	1	0.00	1		0	0	0	0.06	1
harp2	141.11	161 634	0.43	74		74	74	74	139.65	161 634
irp	47.22	451	0.14	0		0	0	0	47.03	451
khb05250	0.44	1	0.00	1		1	1	1	0.44	1
l152lav	2.25	54	0.00	0		0	0	0	2.24	54
liu	1 800.00	699 940	0.69	0	time	0	0	0	1 800.00	717 219
lseu	0.22	384	0.00	2		0	0	1	0.22	384
manna81	0.41	1	0.00	0		0	0	0	0.41	1
markshare1	1 669.93	15 888 811	2.34	547		547	547	547	1 461.44	12 359 490
markshare2	1 126.92	9 615 970	1.63	532		532	532	532	1 573.05	12 642 384
markshare2_1	1 800.00	21 108 996	2.91	575	time	575	575	575	1 800.00	20 914 071
markshare4_0	149.50	1 978 313	0.27	562		562	562	562	157.26	2 016 058
mas74	619.84	3 368 786	0.52	0		0	0	0	619.42	3 368 786
mas76	55.87	330 335	0.06	0		0	0	0	55.79	330 335
mas284	10.43	15 979	0.03	0		0	0	0	10.38	15 979
mik.250-20-75.1	3.62	9 089	0.11	244		244	244	244	3.24	9 089
mik.250-20-75.2	2.61	5 031	0.10	203		203	203	203	2.40	5 031
mik.250-20-75.3	2.61	4 597	0.06	206		206	206	206	2.43	4 597
mik.250-20-75.4	28.41	103 731	0.17	156		156	156	156	27.30	102 249
mik.250-20-75.5	4.82	13 349	0.16	267		267	267	267	4.59	13 349

continued on next page

Problem Name	Time/sec.	Nodes	ZI Time/sec.	Solutions ZI-1	Time limit	ZI1	ZI2	ZI5	Default: Time/sec.	Nodes
misc03	0.97	67	0.00	1		1	1	1	0.97	67
misc06	0.26	5	0.00	4		4	4	4	0.26	5
misc07	16.57	26 110	0.09	0		0	0	0	16.41	26 110
mitre	6.00	1	0.00	0		0	0	0	6.00	1
mkc	1 800.00	695 684	37.65	0	time	0	0	0	1 800.00	712 841
mkc1	1 800.00	639 244	58.39	0	time	0	0	0	1 800.00	661 638
mod008	0.66	343	0.01	126		126	126	126	0.66	343
mod010	1.05	1	0.00	0		0	0	0	1.05	1
mod011	63.83	1 899	0.17	46		41	41	46	64.96	1 951
modglob	0.61	84	0.00	58		58	58	58	0.60	84
momentum1	1 800.00	2 230	1.01	0	time	0	0	0	1 800.00	2 226
momentum2	1 800.07	4 100	2.30	0	time	0	0	0	1 800.01	4 101
momentum3	1 800.00	1	0.00	0	time	0	0	0	1 800.00	1
msc98-ip	1 800.01	24	0.05	0	time	0	0	0	1 800.00	24
mzzv11	282.55	2 559	0.36	0		0	0	0	282.43	2 559
mzzv42z	146.12	714	0.16	0		0	0	0	145.95	714
neos1	1.58	1	0.00	0		0	0	0	1.57	1
neos2	45.54	16 448	0.05	0		0	0	0	45.52	16 448
neos3	1 618.98	372 284	2.75	0		0	0	0	1 615.90	372 284
neos4	2.48	1	0.00	0		0	0	0	2.49	1
neos5	1 145.48	6 015 968	1.16	266		266	266	266	1 172.08	6 091 898
neos6	111.90	2 069	0.18	0		0	0	0	111.74	2 069
neos7	65.44	45 256	0.08	0		0	0	0	65.67	45 256
neos8	14.35	1	0.00	0		0	0	0	14.40	1
neos9	1 800.01	1 866	4.82	0	time	0	0	0	1 800.00	1 902
neos10	20.64	9	0.00	0		0	0	0	20.69	9
neos11	226.25	2 703	0.20	0		0	0	0	225.91	2 703
neos12	1 104.44	2 406	0.52	0		0	0	0	1 104.92	2 406
neos13	185.81	1 601	0.99	0		0	0	0	185.29	1 601
neos16	1 800.00	1 109 208	0.50	0	time	0	0	0	1 800.00	1 114 409
neos20	6.08	583	0.01	3		2	2	3	6.50	600
neos21	27.27	2 117	0.27	283		16	16	283	27.01	2 117
neos22	0.64	1	0.00	0		0	0	0	0.64	1
neos23	8.61	8 549	0.15	0		0	0	0	8.48	8 549
neos616206	1 800.00	984 437	12.43	0	time	0	0	0	1 800.00	989 497

continued on next page

Problem Name	Time/sec.	Nodes	ZI Time/sec.	Solutions ZI-1	Time limit	ZI1	ZI2	ZI5	Default: Time/sec.	Nodes
neos632659	61.69	170 888	0.05	17		16	16	17	62.49	172 218
neos648910	1.12	38	0.00	0		0	0	0	1.13	38
neos808444	1 800.01	155	0.19	0	time	0	0	0	1 800.04	151
neos818918	1 800.00	617 767	0.13	0	time	0	0	0	1 800.00	618 699
neos823206	587.40	22 488	0.38	0		0	0	0	587.16	22 488
neos897005	33.68	1	0.00	0		0	0	0	33.84	1
net12	1 800.01	3 418	1.18	0	time	0	0	0	1 800.00	3 425
noswot	255.91	660 898	0.11	5		0	0	0	229.34	611 086
ns1648184	1 800.00	301 981	2.30	0	time	0	0	0	1 800.00	301 871
ns1688347	456.64	8 335	0.33	0		0	0	0	456.30	8 335
ns1671066	1 800.00	690 897	5.79	0	time	0	0	0	1 800.00	693 117
ns1692855	1 800.00	10 766	0.81	0	time	0	0	0	1 800.00	10 758
nug08	130.76	1	0.00	0		0	0	0	130.55	1
nw04	96.07	5	0.00	0		0	0	0	95.11	5
opt1217	0.33	1	0.00	0		0	0	0	0.33	1
p0033	0.02	1	0.00	0		0	0	0	0.02	1
p0201	0.80	68	0.00	5		5	5	5	0.80	68
p0282	0.43	10	0.00	5		2	2	5	0.43	10
p0548	0.21	68	0.00	1		1	1	1	0.21	68
p2756	1.79	203	0.01	25		34	34	25	1.83	208
pk1	55.40	247 509	0.39	0		0	0	0	54.95	247 509
pp08a	1.04	573	0.03	126		126	126	126	1.04	573
pp08aCUTS	0.98	73	0.00	40		40	40	40	0.97	73
prod1	14.80	24 615	0.18	0		0	0	0	14.64	24 615
prod2	43.57	49 024	0.49	0		0	0	0	43.15	49 024
protfold	1 800.00	3 918	0.29	0	time	0	0	0	1 800.00	3 913
qap10	214.97	1	0.00	0		0	0	0	215.49	1
qiu	59.58	12 750	0.37	434		434	434	434	59.26	12 750
qnet1	1.75	13	0.00	0		0	0	0	1.76	13
qnet1_o	1.71	28	0.00	1		1	1	1	1.70	28
ran8x32	14.09	16 190	0.32	336		336	336	336	11.68	13 093
ran10x26	32.89	40 074	0.57	579		579	579	579	29.02	37 371
ran12x21	72.24	87 581	0.44	446		446	446	446	64.63	86 000
ran13x13	29.40	39 415	0.43	577		577	577	577	28.23	38 316
rd-rplusc-21	1 800.01	40 351	4.30	0	time	0	0	0	1 800.00	40 567

continued on next page

Problem Name	Time/sec.	Nodes	ZI Time/sec.	Solutions ZI-1	Time limit	ZI1	ZI2	ZI5	Default: Time/sec.	Nodes
rentacar	1.69	9	0.00	0		0	0	0	1.68	9
rgn	0.23	1	0.00	0		0	0	0	0.24	1
roll3000	1 800.00	454 485	14.04	0	time	0	0	0	1 800.00	459 018
rout	34.81	24 218	0.17	0		0	0	0	34.59	24 218
set1ch	0.53	21	0.00	14		14	14	14	0.53	21
seymour	1 800.00	32 748	9.13	166	time	166	166	166	1 800.00	33 466
seymour1	236.87	4 626	0.87	367		367	367	367	235.74	4 626
sp97ar	1 800.01	9 990	11.90	18	time	0	0	5	1 800.01	9 859
stein27	0.81	3 963	0.01	196		196	196	196	0.76	4 131
stein45	15.66	52 789	1.36	442		442	442	442	14.85	52 507
stp3d	1 800.42	2	0.00	0	time	0	0	0	1 803.85	2
swath1	26.75	741	0.01	0		0	0	0	26.65	741
swath2	43.27	2 125	0.02	0		0	0	0	43.14	2 125
swath3	193.02	18 239	0.16	0		0	0	0	193.38	18 239
t1717	1 800.03	622	0.48	0	time	0	0	0	1 800.03	625
timtab1	635.67	941 228	0.13	0		0	0	0	633.74	941 228
timtab2	1 800.00	1 513 196	0.35	0	time	0	0	0	1 800.00	1 518 471
tr12-30	1 659.45	1 155 662	0.18	0		0	0	0	1 659.66	1 155 662
vpm1	0.03	1	0.00	0		0	0	0	0.02	1
vpm2	0.91	399	0.01	7		6	6	7	0.89	399

Table 6.4: Results of the second ZI round Experiment:

Problem Name	γ = 0.0 Sols	T	t_{opt}	Δ	γ = 0.1 Sols	T	t_{opt}	Δ	γ = 0.25 Sols	T	t_{opt}	Δ
10teams	0	9.39	9.39	0.0000	0	8.78	8.78	0.0000	0	8.76	8.75	0.0000
30:70:4_5:0_5:100	2	177.96	167.19	0.0000	2	134.05	134.01	0.0000	2	133.33	133.29	0.0000
30:70:4_5:0_95:98	3	141.55	141.54	0.0000	2	98.27	98.27	0.0000	2	98.41	98.40	0.0000
30:70:4_5:0_95:100	2	170.93	159.22	0.0000	2	125.51	125.46	0.0000	2	125.70	125.66	0.0000
a1c1s1	2	1 800.00	835.33	1.1697	1	1 800.00	281.33	1.6667	1	1 800.00	986.55	1.0513
acc-2	0	69.34	69.27	0.0000	0	69.34	69.32	0.0000	0	69.37	69.36	0.0000
acc-4	0	116.13	116.07	0.0000	0	116.03	116.02	0.0000	0	116.03	116.02	0.0000
aflow30a	0	12.05	8.23	0.0000	0	11.97	8.18	0.0000	0	12.02	8.22	0.0000
aflow40b	0	1 515.25	447.80	0.0000	0	1 518.08	451.04	0.0000	0	1 528.21	452.88	0.0000
air03	0	29.19	29.19	0.0000	0	28.89	28.88	0.0000	0	29.04	29.04	0.0000
air04	0	71.79	69.46	0.0000	0	71.99	69.66	0.0000	0	48.17	47.45	0.0000
air05	0	51.98	49.19	0.0000	0	35.24	33.39	0.0000	0	25.89	24.56	0.0000
arki001	0	1 441.81	1 430.79	0.0000	0	1 438.75	1 427.77	0.0000	0	1 437.85	1 426.81	0.0000
atlanta-ip	2	1 800.01	853.97	9.9988	0	1 800.01	1 783.06	9.9988	0	1 800.06	1 723.10	9.9988
bc1	0	211.47	176.42	0.0000	0	210.58	175.74	0.0000	0	211.14	176.19	0.0000
bell3a	0	10.20	0.03	0.0000	0	10.22	0.03	0.0000	0	10.29	0.02	0.0000
bell5	0	0.36	0.12	0.0000	0	0.35	0.11	0.0000	0	0.34	0.11	0.0000
bienst1	0	25.27	13.10	0.0000	0	25.40	13.16	0.0000	0	25.48	13.19	0.0000
bienst2	0	179.13	100.15	0.0000	0	179.49	100.44	0.0000	0	179.39	100.48	0.0000
binkar10_1	0	199.62	148.01	0.0000	0	198.94	146.87	0.0000	0	198.42	147.15	0.0000
cap6000	0	2.64	2.30	0.0000	0	2.63	2.29	0.0000	0	2.64	2.31	0.0000
dano3_3	0	124.14	124.14	0.0000	0	124.07	124.07	0.0000	0	124.07	124.07	0.0000
dano3_4	0	154.82	138.82	0.0000	0	154.47	138.43	0.0000	0	155.11	139.16	0.0000
dano3_5	0	315.87	303.50	0.0000	0	315.68	303.36	0.0000	0	314.97	302.66	0.0000
dano3mip	0	1 800.00	1 518.82	100.0000	0	1 800.00	1 519.10	100.0000	0	1 800.00	1 519.06	100.0000
danoint	0	1 800.00	29.31	0.0000	0	1 800.00	29.41	0.0000	0	1 800.00	29.39	0.0000
dcmulti	0	0.90	0.87	0.0000	0	0.91	0.87	0.0000	0	0.90	0.87	0.0000
egout	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000
eilD76	0	18.81	18.81	0.0000	0	18.22	18.22	0.0000	0	18.21	18.21	0.0000
fast0507	0	1 922.87	1 491.54	0.0000	0	299.43	153.92	0.0000	0	267.94	130.67	0.0000
fiber	1	1.04	0.70	0.0000	1	0.94	0.61	0.0000	1	0.94	0.61	0.0000
fixnet6	0	1.81	1.79	0.0000	0	1.80	1.78	0.0000	0	1.81	1.79	0.0000
gen	0	0.06	0.06	0.0000	0	0.07	0.07	0.0000	0	0.07	0.07	0.0000

continued on next page

Problem Name	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ
gesa2-o	0	1.03	1.03	0.0000	0	1.03	1.03	0.0000	0	1.02	1.02	0.0000
gesa2	0	1.17	1.17	0.0000	0	1.16	1.16	0.0000	0	1.18	1.18	0.0000
gesa3	0	1.12	1.12	0.0000	0	1.13	1.13	0.0000	0	1.13	1.13	0.0000
glass4	0	1 800.00	1 753.82	0.0000	0	1 800.00	1 746.13	0.0000	0	1 800.00	1 754.66	0.0000
gt2	1	0.08	0.07	0.0000	0	0.08	0.07	0.0000	0	0.07	0.06	0.0000
harp2	3	244.08	189.78	0.0000	1	253.29	217.44	0.0000	1	254.24	217.99	0.0000
irp	0	322.73	319.66	0.0000	0	96.32	95.20	0.0000	0	63.87	62.98	0.0000
khb05250	0	0.45	0.36	0.0000	0	0.44	0.36	0.0000	0	0.44	0.36	0.0000
l152lav	0	3.98	3.79	0.0000	0	3.97	3.78	0.0000	0	2.73	2.60	0.0000
liu	0	1 800.00	831.85	100.0000	0	1 800.00	823.96	100.0000	0	1 800.00	827.03	100.0000
lseu	0	0.21	0.20	0.0000	0	0.23	0.21	0.0000	0	0.22	0.20	0.0000
manna81	0	1.27	1.27	0.0000	0	0.41	0.41	0.0000	0	0.40	0.40	0.0000
markshare1	2	1 564.52	293.30	600.0000	2	1 561.40	293.53	600.0000	2	1 558.76	292.25	600.0000
markshare2	2	1 614.36	314.35	1 700.0000	2	1 616.25	314.92	1 700.0000	2	1 619.27	315.41	1 700.0000
markshare2_1	1	1 800.00	1 787.68	∞	1	1 800.00	1 798.28	∞	1	1 800.00	1 796.80	∞
markshare4_0	1	124.19	29.51	0.0000	1	123.91	29.37	0.0000	1	124.13	29.54	0.0000
mas74	0	622.88	107.97	0.0000	0	624.09	108.17	0.0000	0	624.42	108.18	0.0000
mas76	0	56.37	0.52	0.0000	0	56.29	0.53	0.0000	0	56.09	0.54	0.0000
mas284	0	10.49	6.06	0.0000	0	10.47	6.01	0.0000	0	10.48	6.05	0.0000
mik.250-20-75.1	0	3.30	0.69	0.0000	0	3.29	0.70	0.0000	0	3.31	0.70	0.0000
mik.250-20-75.2	0	2.44	0.77	0.0000	0	2.46	0.78	0.0000	0	2.45	0.77	0.0000
mik.250-20-75.3	0	2.51	0.83	0.0000	0	2.49	0.82	0.0000	0	2.49	0.83	0.0000
mik.250-20-75.4	0	27.38	0.82	0.0000	0	27.54	0.83	0.0000	0	27.51	0.82	0.0000
mik.250-20-75.5	0	4.67	0.98	0.0000	0	4.66	0.99	0.0000	0	4.67	0.98	0.0000
misc03	0	0.99	0.53	0.0000	0	0.98	0.53	0.0000	0	0.98	0.53	0.0000
misc06	0	0.26	0.26	0.0000	0	0.27	0.27	0.0000	0	0.26	0.26	0.0000
misc07	0	16.73	0.68	0.0000	0	16.73	0.68	0.0000	0	16.68	0.65	0.0000
mitre	1	6.25	6.12	0.0000	0	6.22	6.22	0.0000	0	6.14	6.14	0.0000
mkc	0	1 800.00	797.88	1.7086	0	1 800.00	781.21	1.7086	0	1 800.00	793.95	1.7086
mkc1	0	1 800.00	6.73	0.0000	0	1 800.00	5.87	0.0000	0	1 800.00	5.94	0.0000
mod008	0	0.69	0.04	0.0000	0	0.65	0.03	0.0000	0	0.66	0.03	0.0000
mod010	0	1.41	1.41	0.0000	0	1.42	1.42	0.0000	0	1.44	1.44	0.0000
mod011	1	66.89	55.50	0.0000	1	67.15	55.76	0.0000	1	67.13	55.61	0.0000
modglob	0	0.61	0.60	0.0000	0	0.60	0.60	0.0000	0	0.61	0.61	0.0000
momentum2	0	1 800.01	1 131.10	25.1889	0	1 800.02	1 120.82	25.1889	0	1 800.00	1 123.97	25.1889

continued on next page

Problem Name	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ
msc98-ip	0	1 800.01	1 582.11	44.4810	0	1 800.37	1 586.63	44.4810	0	1 800.98	1 583.02	44.4810
mzzv11	2	299.40	266.62	0.0000	0	286.28	267.62	0.0000	0	285.44	266.83	0.0000
mzzv42z	0	153.29	150.58	0.0000	0	148.10	145.41	0.0000	0	148.01	145.28	0.0000
neos2	0	47.09	46.77	0.0000	0	47.24	46.90	0.0000	0	46.82	46.48	0.0000
neos3	0	1 654.37	1 627.26	0.0000	0	1 651.03	1 623.92	0.0000	0	1 639.77	1 612.79	0.0000
neos5	0	1 348.45	167.82	0.0000	0	1 345.09	167.74	0.0000	0	1 346.30	167.91	0.0000
neos6	0	152.75	152.74	0.0000	0	115.00	115.00	0.0000	0	114.56	114.55	0.0000
neos7	0	67.42	1.84	0.0000	0	66.74	1.83	0.0000	0	66.84	1.84	0.0000
neos8	0	14.55	14.55	0.0000	0	14.47	14.47	0.0000	0	14.48	14.48	0.0000
neos9	0	1 800.00	404.08	0.0000	0	1 800.01	400.63	0.0000	0	1 800.00	406.31	0.0000
neos10	0	20.91	20.91	0.0000	0	20.86	20.86	0.0000	0	21.27	21.27	0.0000
neos13	0	188.51	154.17	0.0000	0	188.95	154.76	0.0000	0	188.60	154.48	0.0000
neos16	0	1 800.00	326.10	100.0000	0	1 800.00	326.25	100.0000	0	1 800.00	326.18	100.0000
neos21	0	27.17	12.41	0.0000	0	27.23	12.42	0.0000	0	27.23	12.46	0.0000
neos22	0	0.67	0.66	0.0000	0	0.64	0.64	0.0000	0	0.64	0.64	0.0000
neos23	0	8.67	2.53	0.0000	0	8.69	2.52	0.0000	0	8.80	2.53	0.0000
neos616206	0	1 800.00	90.47	0.0000	0	1 800.00	90.44	0.0000	0	1 800.00	90.30	0.0000
neos632659	0	62.68	23.10	0.0000	0	62.51	22.99	0.0000	0	62.62	23.03	0.0000
neos648910	0	1.12	0.31	0.0000	0	1.13	0.31	0.0000	0	1.12	0.31	0.0000
neos818918	0	1 800.00	21.35	0.0000	0	1 800.00	21.35	0.0000	0	1 800.00	21.42	0.0000
neos823206	0	589.86	156.48	0.0000	0	588.84	156.35	0.0000	0	589.46	156.10	0.0000
net12	0	1 800.01	1 002.02	0.0000	0	1 800.01	1 001.41	0.0000	0	1 800.00	999.78	0.0000
noswot	0	182.57	35.89	0.0000	0	184.14	36.02	0.0000	0	182.53	35.88	0.0000
ns1648184	0	1 800.00	1 119.84	0.0110	0	1 800.00	1 120.68	0.0110	0	1 800.00	1 119.62	0.0110
ns1688347	1	511.02	511.02	0.0000	0	459.98	459.98	0.0000	1	511.01	511.01	0.0000
ns1671066	6	1 800.00	158.65	0.1901	6	1 800.00	155.52	0.1901	6	1 800.00	154.35	0.1901
ns1692855	0	1 800.03	456.58	22.2222	0	1 800.03	456.97	22.2222	0	1 800.03	456.86	22.2222
nug08	0	129.68	129.68	0.0000	0	129.50	129.50	0.0000	0	129.51	129.51	0.0000
nw04	0	1 044.12	995.09	0.0000	0	183.15	173.67	0.0000	0	118.12	112.67	0.0000
opt1217	0	0.33	0.03	0.0000	0	0.33	0.03	0.0000	0	0.33	0.03	0.0000
p0033	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000
p0201	3	0.84	0.66	0.0000	3	0.82	0.65	0.0000	3	0.81	0.63	0.0000
p0282	1	0.43	0.43	0.0000	1	0.43	0.43	0.0000	1	0.43	0.43	0.0000
p0548	2	0.23	0.23	0.0000	2	0.20	0.20	0.0000	2	0.20	0.20	0.0000
p2756	2	1.96	1.96	0.0000	2	1.58	1.58	0.0000	2	1.56	1.56	0.0000

continued on next page

Problem Name	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ
pk1	0	55.64	15.72	0.0000	0	55.71	15.76	0.0000	0	55.70	15.74	0.0000
pp08a	1	1.05	0.89	0.0000	0	1.05	0.89	0.0000	0	1.05	0.88	0.0000
pp08aCUTS	1	1.01	0.54	0.0000	0	0.97	0.51	0.0000	0	0.96	0.50	0.0000
prod1	0	14.99	2.04	0.0000	0	15.02	2.05	0.0000	0	14.86	2.05	0.0000
prod2	0	44.00	42.01	0.0000	0	44.20	42.21	0.0000	0	44.21	42.21	0.0000
protfold	0	1 800.00	649.63	32.2581	0	1 800.00	649.51	32.2581	0	1 800.02	649.94	32.2581
qap10	0	215.76	186.54	0.0000	0	216.08	186.98	0.0000	0	214.78	185.58	0.0000
qiu	1	60.88	7.90	0.0000	0	59.66	6.38	0.0000	0	59.45	6.39	0.0000
qnet1	0	1.89	1.84	0.0000	0	1.84	1.79	0.0000	0	1.80	1.77	0.0000
qnet1_o	0	2.05	2.05	0.0000	0	1.72	1.72	0.0000	0	1.73	1.73	0.0000
ran8x32	0	10.52	4.43	0.0000	0	10.45	4.40	0.0000	0	10.50	4.42	0.0000
ran10x26	0	31.48	17.28	0.0000	0	31.49	17.28	0.0000	0	31.51	17.27	0.0000
ran12x21	0	65.01	12.00	0.0000	0	64.93	11.98	0.0000	0	64.96	12.02	0.0000
ran13x13	0	32.31	19.77	0.0000	0	32.31	19.78	0.0000	0	32.52	19.86	0.0000
rd-rplusc-21	0	1 800.02	1 349.93	1.3201	0	1 800.01	1 344.54	1.3201	0	1 800.01	1 350.51	1.3201
rgn	0	0.23	0.17	0.0000	0	0.23	0.17	0.0000	0	0.23	0.16	0.0000
roll3000	0	1 800.00	418.70	0.2327	0	1 800.00	418.72	0.2327	0	1 800.00	417.16	0.2327
rout	0	35.07	19.17	0.0000	0	35.13	19.21	0.0000	0	35.07	19.12	0.0000
set1ch	0	0.53	0.53	0.0000	0	0.52	0.52	0.0000	0	0.54	0.53	0.0000
seymour	0	1 800.00	1 642.97	0.7092	0	1 800.00	1 640.79	0.7092	0	1 800.00	1 646.57	0.7092
seymour1	0	238.18	52.70	0.0000	0	237.64	52.51	0.0000	0	237.39	52.34	0.0000
sp97ar	3	1 800.00	130.05	2.0153	4	1 800.01	999.76	2.8422	2	1 800.02	288.44	3.3969
stein27	0	0.82	0.16	0.0000	0	0.82	0.17	0.0000	0	0.82	0.17	0.0000
stein45	0	15.81	1.08	0.0000	0	15.81	1.08	0.0000	0	15.79	1.09	0.0000
swath1	0	27.97	27.70	0.0000	0	28.32	28.04	0.0000	0	27.22	26.94	0.0000
t1717	0	1 961.97	1 299.76	100.0000	0	1 800.04	646.13	100.0000	0	1 800.31	640.86	100.0000
timtab1	0	642.69	59.64	0.0000	0	640.51	59.47	0.0000	0	643.46	59.54	0.0000
timtab2	0	1 800.00	1 624.75	6.0807	0	1 800.00	1 618.94	6.0807	0	1 800.00	1 623.07	6.0807
tr12-30	0	1 800.00	229.20	0.0000	0	1 800.00	229.85	0.0000	0	1 800.00	229.14	0.0000
vpm2	0	0.91	0.52	0.0000	0	0.90	0.51	0.0000	0	0.90	0.52	0.0000

Table 6.5: 2-opt results for $\gamma = 0.0, 0.1, 0.25$

Problem Name	$\gamma = 0.4$				$\gamma = 0.6$				$\gamma = 0.75$			
	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ
10teams	0	8.70	8.69	0.0000	0	8.81	8.81	0.0000	0	8.72	8.72	0.0000
30:70:4_5:0_5:100	1	135.71	135.69	0.0000	1	136.25	136.22	0.0000	1	135.52	135.51	0.0000
30:70:4_5:0_95:98	2	97.15	97.14	0.0000	1	98.10	98.10	0.0000	2	96.87	96.87	0.0000
30:70:4_5:0_95:100	2	125.59	125.55	0.0000	1	127.74	127.72	0.0000	1	125.40	125.37	0.0000
a1c1s1	1	1 800.00	283.55	1.6667	0	1 800.00	245.57	0.5093	0	1 800.00	242.02	0.5093
acc-2	0	69.23	69.23	0.0000	0	69.55	69.54	0.0000	0	69.26	69.25	0.0000
acc-4	0	116.14	116.13	0.0000	0	116.71	116.70	0.0000	0	116.06	116.05	0.0000
aflow30a	0	12.01	8.17	0.0000	0	12.08	8.20	0.0000	0	12.00	8.19	0.0000
aflow40b	0	1 526.63	451.27	0.0000	0	1 552.17	459.72	0.0000	0	1 505.75	446.05	0.0000
air03	0	28.89	28.88	0.0000	0	29.55	29.55	0.0000	0	29.00	28.99	0.0000
air04	0	47.63	46.95	0.0000	0	47.84	47.17	0.0000	0	47.38	46.72	0.0000
air05	0	24.02	22.82	0.0000	0	24.12	22.91	0.0000	0	23.65	22.44	0.0000
arki001	0	1 441.29	1 430.23	0.0000	0	1 462.79	1 451.62	0.0000	0	1 455.61	1 444.49	0.0000
atlanta-ip	0	1 800.35	1 716.12	9.9988	0	1 800.01	1 731.60	9.9988	0	1 800.56	1 747.27	9.9988
bc1	0	211.84	176.79	0.0000	0	213.04	177.70	0.0000	0	211.83	176.79	0.0000
bell3a	0	10.20	0.03	0.0000	0	10.27	0.02	0.0000	0	10.26	0.03	0.0000
bell5	0	0.36	0.12	0.0000	0	0.35	0.12	0.0000	0	0.36	0.12	0.0000
bienst1	0	25.36	13.13	0.0000	0	25.47	13.22	0.0000	0	25.39	13.17	0.0000
bienst2	0	179.30	100.39	0.0000	0	179.75	100.55	0.0000	0	180.00	100.79	0.0000
binkar10_1	0	199.31	147.27	0.0000	0	200.94	148.61	0.0000	0	201.82	148.39	0.0000
cap6000	0	2.63	2.30	0.0000	0	2.61	2.28	0.0000	0	2.63	2.30	0.0000
dano3_3	0	123.84	123.84	0.0000	0	124.52	124.51	0.0000	0	125.80	125.79	0.0000
dano3_4	0	154.56	138.61	0.0000	0	155.27	139.22	0.0000	0	156.48	140.31	0.0000
dano3_5	0	315.32	302.93	0.0000	0	316.67	304.29	0.0000	0	316.71	304.25	0.0000
dano3mip	0	1 800.01	1 519.70	100.0000	0	1 800.00	1 524.38	100.0000	0	1 800.20	1 530.14	100.0000
danoint	0	1 800.00	29.32	0.0000	0	1 800.00	29.47	0.0000	0	1 800.00	29.56	0.0000
dcmulti	0	0.92	0.88	0.0000	0	0.91	0.88	0.0000	0	0.91	0.87	0.0000
egout	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000	0	0.02	0.02	0.0000
eilD76	0	18.25	18.25	0.0000	0	18.35	18.35	0.0000	0	18.28	18.28	0.0000
fast0507	0	267.82	129.07	0.0000	0	267.42	129.29	0.0000	0	265.13	127.46	0.0000
fiber	1	0.95	0.61	0.0000	1	0.93	0.60	0.0000	1	0.94	0.60	0.0000
fixnet6	0	1.80	1.79	0.0000	0	1.79	1.78	0.0000	0	1.80	1.78	0.0000
gen	0	0.07	0.07	0.0000	0	0.07	0.07	0.0000	0	0.08	0.08	0.0000

continued on next page

Problem Name	γ = 0.4 Sols	T	t_{opt}	Δ	γ = 0.6 Sols	T	t_{opt}	Δ	γ = 0.75 Sols	T	t_{opt}	Δ
gesa2-o	0	1.02	1.02	0.0000	0	1.04	1.04	0.0000	0	1.03	1.03	0.0000
gesa2	0	1.16	1.16	0.0000	0	1.16	1.16	0.0000	0	1.16	1.16	0.0000
gesa3	0	1.12	1.11	0.0000	0	1.12	1.12	0.0000	0	1.13	1.13	0.0000
glass4	0	1 800.00	1 749.67	0.0000	0	1 800.00	1 773.48	0.0000	0	1 800.00	1 763.24	0.0000
gt2	0	0.07	0.06	0.0000	0	0.08	0.07	0.0000	0	0.07	0.07	0.0000
harp2	1	208.08	165.78	0.0000	0	144.61	88.03	0.0000	0	142.44	87.01	0.0000
irp	0	58.20	57.36	0.0000	0	51.51	50.74	0.0000	0	49.08	48.36	0.0000
khb05250	0	0.43	0.35	0.0000	0	0.45	0.37	0.0000	0	0.45	0.37	0.0000
l152lav	0	2.32	2.21	0.0000	0	2.30	2.19	0.0000	0	2.32	2.21	0.0000
liu	0	1 800.00	827.98	100.0000	0	1 800.00	839.17	100.0000	0	1 800.00	825.85	100.0000
lseu	0	0.21	0.20	0.0000	0	0.21	0.20	0.0000	0	0.22	0.20	0.0000
manna81	0	0.40	0.40	0.0000	0	0.40	0.40	0.0000	0	0.41	0.41	0.0000
markshare1	2	1 559.58	292.96	600.0000	1	1 800.00	1 519.85	700.0000	1	1 800.00	1 514.93	700.0000
markshare2	2	1 616.58	314.77	1 700.0000	2	1 191.22	633.06	2 200.0000	0	1 246.92	330.95	1 800.0000
markshare2_1	1	1 800.00	1 796.16	∞	1	1 800.00	1 794.73	∞	1	1 800.00	1 793.63	∞
markshare4_0	1	123.43	29.31	0.0000	2	155.29	38.34	0.0000	2	159.90	39.42	0.0000
mas74	0	624.21	108.17	0.0000	0	623.47	108.05	0.0000	0	623.51	108.02	0.0000
mas76	0	56.20	0.54	0.0000	0	56.21	0.53	0.0000	0	56.02	0.53	0.0000
mas284	0	10.44	6.02	0.0000	0	10.45	6.01	0.0000	0	10.55	6.08	0.0000
mik.250-20-75.1	0	3.28	0.69	0.0000	0	3.30	0.70	0.0000	0	3.30	0.71	0.0000
mik.250-20-75.2	0	2.46	0.79	0.0000	0	2.44	0.78	0.0000	0	2.48	0.80	0.0000
mik.250-20-75.3	0	2.51	0.83	0.0000	0	2.52	0.84	0.0000	0	2.51	0.84	0.0000
mik.250-20-75.4	0	27.52	0.82	0.0000	0	27.51	0.83	0.0000	0	27.46	0.83	0.0000
mik.250-20-75.5	0	4.65	0.97	0.0000	0	4.68	0.98	0.0000	0	4.65	0.97	0.0000
misc03	0	0.97	0.52	0.0000	0	0.98	0.53	0.0000	0	0.97	0.52	0.0000
misc06	0	0.27	0.27	0.0000	0	0.27	0.26	0.0000	0	0.26	0.26	0.0000
misc07	0	16.78	0.65	0.0000	0	16.78	0.65	0.0000	0	16.63	0.64	0.0000
mitre	0	6.24	6.24	0.0000	0	6.06	6.06	0.0000	0	6.13	6.13	0.0000
mkc	0	1 800.00	794.17	1.7086	0	1 800.00	788.81	1.7086	0	1 800.00	785.41	1.7086
mkc1	0	1 800.00	5.94	0.0000	0	1 800.00	5.85	0.0000	0	1 800.00	5.89	0.0000
mod008	0	0.67	0.03	0.0000	0	0.66	0.03	0.0000	0	0.66	0.03	0.0000
mod010	0	1.08	1.08	0.0000	0	1.07	1.07	0.0000	0	1.07	1.07	0.0000
mod011	0	60.80	53.33	0.0000	0	60.73	53.41	0.0000	0	60.77	53.38	0.0000
modglob	0	0.61	0.60	0.0000	0	0.60	0.59	0.0000	0	0.60	0.60	0.0000

continued on next page

Problem Name	$\gamma = 0.4$ Sols	T	t_{opt}	Δ	$\gamma = 0.6$ Sols	T	t_{opt}	Δ	$\gamma = 0.75$ Sols	T	t_{opt}	Δ
momentum2	0	1 800.00	1 125.84	25.1889	0	1 800.02	1 119.22	25.1889	0	1 800.01	1 118.19	25.1889
msc98-ip	0	1 800.02	1 583.27	44.4810	0	1 800.89	1 586.76	44.4810	0	1 800.76	1 583.88	44.4810
mzzv11	0	284.90	266.25	0.0000	0	285.89	267.15	0.0000	0	285.92	267.35	0.0000
mzzv42z	0	147.17	144.48	0.0000	0	148.10	145.40	0.0000	0	147.64	144.94	0.0000
neos2	0	46.86	46.54	0.0000	0	46.77	46.43	0.0000	0	46.80	46.46	0.0000
neos3	0	1 642.31	1 615.27	0.0000	0	1 642.62	1 615.31	0.0000	0	1 647.00	1 619.96	0.0000
neos5	0	1 345.64	167.76	0.0000	0	1 343.73	167.76	0.0000	0	1 348.29	167.87	0.0000
neos6	0	113.35	113.35	0.0000	0	113.13	113.13	0.0000	0	113.14	113.14	0.0000
neos7	0	66.70	1.81	0.0000	0	66.21	1.81	0.0000	0	67.19	1.84	0.0000
neos8	0	14.67	14.67	0.0000	0	14.66	14.66	0.0000	0	14.41	14.41	0.0000
neos9	0	1 800.02	402.68	0.0000	0	1 800.01	403.70	0.0000	0	1 800.00	402.46	0.0000
neos10	0	20.86	20.86	0.0000	0	21.20	21.20	0.0000	0	20.82	20.82	0.0000
neos13	0	189.30	155.03	0.0000	0	188.61	154.46	0.0000	0	188.27	154.02	0.0000
neos16	0	1 800.00	328.30	100.0000	0	1 800.00	325.29	100.0000	0	1 800.00	326.13	100.0000
neos21	0	27.28	12.41	0.0000	0	27.21	12.38	0.0000	0	27.20	12.39	0.0000
neos22	0	0.65	0.65	0.0000	0	0.64	0.64	0.0000	0	0.66	0.66	0.0000
neos23	0	8.73	2.53	0.0000	0	8.68	2.52	0.0000	0	8.65	2.53	0.0000
neos616206	0	1 800.00	89.96	0.0000	0	1 800.00	90.31	0.0000	0	1 800.00	89.95	0.0000
neos632659	0	62.65	23.06	0.0000	0	62.81	23.11	0.0000	0	62.48	23.07	0.0000
neos648910	0	1.14	0.31	0.0000	0	1.12	0.31	0.0000	0	1.13	0.31	0.0000
neos818918	0	1 800.00	21.40	0.0000	0	1 800.00	21.36	0.0000	0	1 800.00	21.31	0.0000
neos823206	0	588.92	156.17	0.0000	0	590.42	156.34	0.0000	0	589.68	155.94	0.0000
net12	0	1 800.00	999.71	0.0000	0	1 800.01	1 002.69	0.0000	0	1 800.00	1 000.94	0.0000
noswot	0	182.48	35.93	0.0000	0	182.64	35.95	0.0000	0	183.02	36.11	0.0000
ns1648184	0	1 800.00	1 119.91	0.0110	0	1 800.00	1 117.21	0.0110	0	1 800.00	1 119.02	0.0110
ns1688347	1	510.82	510.82	0.0000	1	510.88	510.88	0.0000	0	458.88	458.87	0.0000
ns1671066	6	1 800.00	154.14	0.1901	6	1 800.00	155.63	0.1901	6	1 800.00	154.83	0.1901
ns1692855	0	1 800.04	457.36	22.2222	0	1 800.00	457.61	22.2222	0	1 800.00	457.09	22.2222
nug08	0	129.75	129.75	0.0000	0	129.60	129.60	0.0000	0	130.20	130.20	0.0000
nw04	0	105.74	101.20	0.0000	0	96.46	92.44	0.0000	0	96.02	91.95	0.0000
opt1217	0	0.33	0.03	0.0000	0	0.34	0.03	0.0000	0	0.34	0.03	0.0000
p0033	0	0.02	0.02	0.0000	0	0.03	0.03	0.0000	0	0.02	0.02	0.0000
p0201	3	0.81	0.63	0.0000	3	0.81	0.64	0.0000	3	0.81	0.63	0.0000
p0282	1	0.43	0.43	0.0000	0	0.43	0.43	0.0000	0	0.44	0.43	0.0000

continued on next page

Problem Name	$\gamma = 0.4$				$\gamma = 0.6$				$\gamma = 0.75$			
	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ	Sols	T	t_{opt}	Δ
p0548	2	0.20	0.20	0.0000	2	0.20	0.20	0.0000	2	0.20	0.20	0.0000
p2756	2	1.56	1.56	0.0000	2	1.86	1.86	0.0000	3	1.75	1.75	0.0000
pk1	0	55.68	15.77	0.0000	0	55.73	15.81	0.0000	0	55.81	15.78	0.0000
pp08a	0	1.04	0.88	0.0000	0	1.05	0.88	0.0000	0	1.06	0.89	0.0000
pp08aCUTS	0	0.97	0.50	0.0000	0	0.97	0.51	0.0000	0	0.97	0.50	0.0000
prod1	0	14.95	2.03	0.0000	0	14.95	2.04	0.0000	0	14.89	2.05	0.0000
prod2	0	43.97	42.02	0.0000	0	43.99	42.03	0.0000	0	44.32	42.34	0.0000
protfold	0	1 800.00	649.13	32.2581	0	1 800.00	649.78	32.2581	0	1 800.00	649.83	32.2581
qap10	0	216.17	186.91	0.0000	0	215.13	185.55	0.0000	0	214.92	185.69	0.0000
qiu	0	59.38	6.37	0.0000	0	59.46	6.39	0.0000	0	59.51	6.36	0.0000
qnet1	0	1.78	1.76	0.0000	0	1.77	1.76	0.0000	0	1.77	1.76	0.0000
qnet1_o	0	1.72	1.72	0.0000	0	1.72	1.72	0.0000	0	1.72	1.72	0.0000
ran8x32	0	10.51	4.43	0.0000	0	10.47	4.41	0.0000	0	10.47	4.41	0.0000
ran10x26	0	31.38	17.23	0.0000	0	31.40	17.21	0.0000	0	31.53	17.29	0.0000
ran12x21	0	65.00	12.03	0.0000	0	64.97	11.97	0.0000	0	64.83	11.97	0.0000
ran13x13	0	32.41	19.80	0.0000	0	32.45	19.84	0.0000	0	32.30	19.77	0.0000
rd-rplusc-21	0	1 800.01	1 357.65	1.3201	0	1 800.00	1 356.94	1.3201	0	1 800.02	1 356.25	1.3201
rgn	0	0.24	0.17	0.0000	0	0.23	0.16	0.0000	0	0.23	0.17	0.0000
roll3000	0	1 800.00	419.46	0.2327	0	1 800.00	417.34	0.2327	0	1 800.00	419.09	0.2327
rout	0	35.10	19.14	0.0000	0	34.95	19.06	0.0000	0	35.19	19.15	0.0000
set1ch	0	0.53	0.52	0.0000	0	0.53	0.52	0.0000	0	0.53	0.52	0.0000
seymour	0	1 800.00	1 638.15	0.7092	0	1 800.00	1 639.42	0.7092	0	1 800.00	1 638.76	0.7092
seymour1	0	238.12	52.55	0.0000	0	238.11	52.46	0.0000	0	238.36	52.61	0.0000
sp97ar	4	1 800.00	760.81	1.8335	5	1 800.00	1 225.29	1.7742	4	1 800.01	347.01	2.3695
stein27	0	0.81	0.17	0.0000	0	0.81	0.17	0.0000	0	0.82	0.17	0.0000
stein45	0	15.88	1.10	0.0000	0	15.79	1.08	0.0000	0	15.79	1.09	0.0000
swath1	0	27.76	27.47	0.0000	0	27.24	26.96	0.0000	0	27.09	26.80	0.0000
t1717	0	1 800.04	646.26	100.0000	0	1 800.03	641.42	100.0000	0	1 800.14	639.63	100.0000
timtab1	0	653.21	60.32	0.0000	0	644.10	59.90	0.0000	0	645.83	59.61	0.0000
timtab2	0	1 800.00	1 636.23	6.0807	0	1 800.00	1 626.89	6.0807	0	1 800.00	1 625.88	6.0807
tr12-30	0	1 800.00	229.75	0.0000	0	1 800.00	229.23	0.0000	0	1 799.84	227.51	0.0000
vpm2	0	0.89	0.51	0.0000	0	0.90	0.52	0.0000	0	0.89	0.51	0.0000

Table 6.6: 2-opt results for $\gamma = 0.4, 0.6, 0.75$

Problem Name	Sols	T	t_{opt}	Δ
10teams	0	8.76	8.76	0
30:70:4_5:0_5:100	0	134.9	134.89	0
30:70:4_5:0_95:98	0	97.55	97.55	0
30:70:4_5:0_95:100	0	126.52	126.5	0
a1c1s1	0	1 800	241.53	0.51
acc-2	0	69.47	69.47	0
acc-4	0	116.63	116.62	0
aflow30a	0	12.04	8.21	0
aflow40b	0	1 523.65	450.95	0
air03	0	29.08	29.07	0
air04	0	47.71	47.04	0
air05	0	23.94	22.74	0
arki001	0	1 450.87	1 439.85	$5.41 \cdot 10^{-8}$
atlanta-ip	0	1 800	1 710.45	10
bc1	0	211.97	176.74	$7.19 \cdot 10^{-8}$
bell3a	0	10.18	$2 \cdot 10^{-2}$	0
bell5	0	0.35	0.12	$1.67 \cdot 10^{-8}$
bienst1	0	25.34	13.14	0
bienst2	0	180.03	100.86	0
binkar10_1	0	200.17	147.98	$5.93 \cdot 10^{-8}$
cap6000	0	2.65	2.31	0
dano3_3	0	124.26	124.26	$5.21 \cdot 10^{-9}$
dano3_4	0	155.24	139.26	$5.03 \cdot 10^{-8}$
dano3_5	0	315.81	303.43	$6.93 \cdot 10^{-9}$
dano3mip	0	1 800	1 527.1	100
danoint	0	1 800	29.61	$5.18 \cdot 10^{-8}$
dcmulti	0	0.92	0.88	0
egout	0	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	0
eilD76	0	18.22	18.22	0
fast0507	0	267.09	128.16	0
fiber	1	0.93	0.6	0
fixnet6	0	1.8	1.78	0
gen	0	$7 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	$2.49 \cdot 10^{-7}$
gesa2-o	0	1.03	1.03	$1.09 \cdot 10^{-7}$
gesa2	0	1.17	1.17	$1.09 \cdot 10^{-7}$
gesa3	0	1.13	1.13	$1.71 \cdot 10^{-7}$
glass4	0	1 800	1 792.57	0
gt2	0	$7 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	0
harp2	0	143.87	87.95	$2.17 \cdot 10^{-5}$
irp	0	48.73	48	$2.96 \cdot 10^{-7}$
khb05250	0	0.45	0.37	0
l152lav	0	2.29	2.18	0
liu	0	1 800	851.78	100
lseu	0	0.22	0.2	0
manna81	0	0.4	0.4	0
markshare1	1	1 800	1 532.67	700
markshare2	0	1 261.15	332.95	1 800
markshare2_1	1	1 800	278.95	$1 \cdot 10^{20}$
markshare4_0	1	148.46	60.26	0
mas74	0	626.19	108.23	0
mas76	0	56.5	0.55	0
mas284	0	10.65	6.2	$1.97 \cdot 10^{-8}$
mik.250-20-75.1	0	3.31	0.7	0
mik.250-20-75.2	0	2.46	0.77	0
mik.250-20-75.3	0	2.54	0.84	0
mik.250-20-75.4	0	27.62	0.83	0

continued on next page

Problem Name	Sols	T	t_{opt}	Δ
mik.250-20-75.5	0	4.67	0.98	0
misc03	0	0.97	0.52	0
misc06	0	0.27	0.27	$2.88 \cdot 10^{-7}$
misc07	0	16.71	0.65	0
mitre	0	6.12	6.12	0
mkc	0	1 800	817.02	1.71
mkc1	0	1 800	5.89	0
mod008	0	0.66	$3 \cdot 10^{-2}$	0
mod010	0	1.07	1.06	0
mod011	0	60.89	53.51	$2.57 \cdot 10^{-8}$
modglob	0	0.6	0.59	$4.82 \cdot 10^{-7}$
momentum2	0	1 800.01	1 123.85	25.19
msc98-ip	0	1 800.42	1 584.87	44.48
mzzv11	0	284.65	265.92	0
mzzv42z	0	148.9	146.16	0
neos2	0	46.71	46.39	$8.79 \cdot 10^{-9}$
neos3	0	1 647.89	1 620.86	0
neos5	0	1 346.01	167.77	0
neos6	0	112.87	112.86	0
neos7	0	66.91	1.86	0
neos8	0	14.56	14.56	0
neos9	0	1 800	399.41	0
neos10	0	20.81	20.81	0
neos13	0	188.83	154.61	$4.29 \cdot 10^{-8}$
neos16	0	1 800	327.34	100
neos21	0	27.17	12.41	0
neos22	0	0.65	0.65	0
neos23	0	8.76	2.53	0
neos616206	0	1 800	90.13	0
neos632659	0	62.7	23.08	0
neos648910	0	1.12	0.3	0
neos818918	0	1 800	21.42	0
neos823206	0	589.78	156.44	$2.01 \cdot 10^{-8}$
net12	0	1 800.01	1 001.68	0
noswot	0	182.92	35.96	0
ns1648184	0	1 800	1 120.3	$1.1 \cdot 10^{-2}$
ns1688347	0	459.14	459.14	0
ns1671066	0	1 800	141.22	$8.22 \cdot 10^{-2}$
ns1692855	0	1 800.03	456.85	22.22
nug08	0	130.19	130.19	0
nw04	0	95.89	91.91	0
opt1217	0	0.33	$3 \cdot 10^{-2}$	0
p0033	0	$3 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	0
p0201	0	0.8	0.57	0
p0282	0	0.43	0.43	0
p0548	3	0.2	0.2	0
p2756	0	1.76	1.76	0
pk1	0	55.86	15.76	0
pp08a	0	1.04	0.88	0
pp08aCUTS	0	0.97	0.5	0
prod1	0	14.95	2.04	0
prod2	0	44.09	42.1	0
protfold	0	1 800	649.18	32.26
qap10	0	215.53	186.34	0
qiu	0	59.3	6.36	$3.76 \cdot 10^{-8}$
qnet1	0	1.75	1.74	$1.19 \cdot 10^{-7}$
qnet1_o	0	1.71	1.71	$1.19 \cdot 10^{-7}$

continued on next page

Problem Name	Sols	T	t_{opt}	Δ
ran8x32	0	10.45	4.39	0
ran10x26	0	31.53	17.27	0
ran12x21	0	64.94	12	0
ran13x13	0	32.44	19.83	0
rd-rplusc-21	0	1 800.01	1 353.72	1.32
rgn	0	0.24	0.17	$2.12 \cdot 10^{-6}$
roll3000	0	1 800	419.91	0.23
rout	0	35.26	19.21	0
set1ch	0	0.53	0.53	0
seymour	0	1 800	1 642.06	0.71
seymour1	0	237.97	52.51	$9.49 \cdot 10^{-8}$
sp97ar	3	1 800	185.29	2.6
stein27	0	0.82	0.18	0
stein45	0	15.84	1.09	0
swath1	0	27.23	26.94	$6.6 \cdot 10^{-8}$
t1717	0	1 800.03	641.59	100
timtab1	0	646.24	59.66	0
timtab2	0	1 800	1 624.04	6.08
tr12-30	0	1 799.45	228.49	0
vpm2	0	0.91	0.52	0

Table 6.7: 2-opt results for $\gamma = 0.95$

	*SCIP*_DEFAULT			SaP_UP		
Problem Name	T	nodes	Time limit	T	nodes	Time limit
10teams	6.09	125		6.03	166	
30:70:4_5:0_5:100	132.55	34		274.16	63	
30:70:4_5:0_95:98	95.06	142		90.18	70	
30:70:4_5:0_95:100	124.11	16		132.55	114	
a1c1s1	1 800.00	84 096	time	1 800.00	121 967	time
acc-0	0.72	1		0.80	1	
acc-1	1.08	1		1.08	1	
acc-2	62.85	110		61.85	92	
acc-3	213.58	206		118.79	177	
acc-4	1 360.97	1 281		1 800.00	4 887	time
acc-5	143.01	258		1 030.18	4 083	
acc-6	54.90	74		56.61	116	
aflow30a	12.54	2 183		13.58	2 622	
aflow40b	1 800.00	206 820	time	1 667.21	331 450	
air03	28.67	1		28.70	1	
air04	63.06	490		52.16	293	
air05	23.37	172		24.51	260	
arki001	809.91	549 918		887.13	577 727	
atlanta-ip	1 800.00	1 359	time	1 800.01	893	time
bc1	185.42	5 512		168.68	4 855	
bell3a	14.19	49 108		9.14	40 143	
bell5	0.35	1 055		0.31	1 163	
bienst1	19.09	17 260		15.90	12 655	
bienst2	146.22	111 563		156.41	114 091	
binkar10_1	198.75	131 082		165.55	109 828	
blend2	0.45	194		0.49	200	
cap6000	2.84	3 767		2.82	3 337	
dano3_3	67.84	7		175.98	10	
dano3_4	206.02	32		137.45	24	
dano3_5	233.07	144		277.14	176	
dano3mip	1 800.01	1 214	time	1 800.00	1 096	time
danoint	1 800.00	324 880	time	1 800.00	343 694	time

continued on next page

	*SCIP*_Default			SaP_Up		
Problem Name	*T*	nodes	Time limit	*T*	nodes	Time limit
dcmulti	1.27	82		1.27	82	
disctom	2.76	1		3.39	1	
ds	1 800.00	263	time	1 800.06	257	time
dsbmip	0.15	1		0.16	1	
egout	0.01	1		0.02	1	
eilD76	18.06	3		18.05	3	
enigma	0.08	115		0.30	998	
fast0507	573.08	1 413		299.85	1 000	
fiber	0.96	11		0.91	17	
fixnet6	1.78	15		2.20	7	
flugpl	0.04	80		0.04	119	
gen	0.07	1		0.07	1	
gesa2-o	1.01	1		1.06	3	
gesa2	1.16	11		1.35	10	
gesa3	1.12	17		1.15	17	
gesa3_o	1.63	11		1.38	11	
glass4	1 800.00	2 977 340	time	1 225.75	1 917 933	
gt2	0.06	1		0.08	1	
harp2	204.81	301 915		258.75	405 711	
irp	19.83	47		46.55	451	
khb05250	0.62	1		0.43	3	
l152lav	2.29	63		1.98	49	
liu	1 800.00	562 928	time	1 800.00	475 720	time
lseu	0.22	384		0.18	277	
manna81	0.40	1		0.50	1	
markshare1	1 800.00	17 056 591	time	1 800.00	16 786 791	time
markshare2	1 236.96	10 237 514		1 800.00	15 182 504	time
markshare2_1	1 800.00	20 935 829	time	1 800.00	21 306 880	time
markshare4_0	151.82	1 978 280		177.08	2 243 694	
mas74	662.81	3 552 279		617.32	3 245 820	
mas76	62.80	363 807		59.76	348 942	
mas284	10.10	15 839		10.51	16 403	
mik.250-20-75.1	3.29	9 089		3.22	8 541	
mik.250-20-75.2	2.55	5 031		2.38	4 145	
mik.250-20-75.3	2.50	4 597		2.87	5 408	
mik.250-20-75.4	27.40	102 249		18.29	68 989	
mik.250-20-75.5	4.71	13 411		4.68	13 503	
misc03	0.91	57		1.05	93	
misc06	0.29	5		0.28	7	
misc07	19.86	32 148		8.51	11 283	
mitre	6.05	1		6.05	1	
mkc	1 437.05	521 696		1 800.00	898 492	time
mkc1	1 800.00	643 917	time	1 800.00	668 171	time
mod008	0.65	343		0.70	369	
mod010	1.04	1		1.08	1	
mod011	63.64	1 899		71.75	2 213	
modglob	0.68	84		0.68	80	
momentum1	1 800.01	1 807	time	1 800.01	1 886	time
momentum2	1 800.01	3 785	time	1 800.01	3 980	time
momentum3	1 800.04	1	time	1 800.04	1	time
msc98-ip	1 800.01	108	time	1 800.48	93	time
mzzv11	282.24	2 559		291.42	2 585	
mzzv42z	145.62	714		146.12	762	
neos1	1.57	1		1.60	1	
neos2	56.90	20 515		44.37	16 094	
neos3	1 442.65	381 491		1 280.90	299 901	

continued on next page

Problem Name	SCIP_Default T	nodes	Time limit	SaP_Up T	nodes	Time limit
neos4	2.48	1		2.46	1	
neos5	1 338.72	6 949 722		1 119.54	5 685 634	
neos6	177.09	2 562		309.90	9 289	
neos7	11.08	9 072		53.26	38 076	
neos8	14.35	1		14.32	1	
neos9	1 800.01	4 849	time	1 800.01	1 901	time
neos10	20.71	9		20.98	9	
neos11	411.88	6 879		232.60	2 584	
neos12	1 055.71	2 457		1 102.32	2 632	
neos13	336.03	5 234		216.63	1 939	
neos16	1 800.00	1 107 341	time	1 800.00	1 154 108	time
neos20	5.65	603		5.48	631	
neos21	27.86	2 056		28.79	2 699	
neos22	1.35	1		0.64	1	
neos23	12.05	10 138		4.97	3 226	
neos616206	1 800.00	1 125 363	time	1 800.00	1 047 234	time
neos632659	410.93	1 675 849		3.25	6 734	
neos648910	2.30	59		0.95	55	
neos808444	1 481.71	553		1 188.54	463	
neos818918	1 800.00	597 975	time	1 800.00	619 562	time
neos823206	532.89	26 863		632.62	26 015	
neos897005	106.99	3		33.44	1	
net12	1 800.00	4 184	time	1 800.00	3 645	time
noswot	231.58	617 209		267.00	652 918	
ns1648184	1 800.00	337 575	time	1 800.00	365 517	time
ns1688347	542.96	17 119		1 522.76	5 317	
ns1671066	99.57	72 386		0.77	3	
ns1692855	1 800.01	10 565	time	1 625.68	12 610	
nug08	150.52	1		131.00	1	
nw04	90.86	6		273.15	5	
opt1217	0.34	1		0.36	1	
p0033	0.02	1		0.02	1	
p0201	0.87	96		0.98	129	
p0282	0.44	10		0.48	1	
p0548	0.20	47		0.19	24	
p2756	1.87	292		1.68	107	
pk1	75.89	343 639		48.95	217 091	
pp08a	0.84	116		1.06	573	
pp08aCUTS	1.06	109		0.98	73	
prod1	15.16	23 587		14.45	22 613	
prod2	84.68	97 573		96.38	118 898	
protfold	1 800.00	1 799	time	1 800.00	3 867	time
qap10	205.22	4		214.67	1	
qiu	71.83	14 826		43.51	9 237	
qnet1	3.35	69		2.15	14	
qnet1_o	1.95	38		1.01	1	
ran8x32	15.12	17 698		15.28	11 692	
ran10x26	27.67	33 765		23.27	23 319	
ran12x21	65.18	80 972		77.14	94 603	
ran13x13	29.98	41 683		24.97	32 351	
rd-rplusc-21	1 800.01	41 625	time	1 800.00	51 885	time
rentacar	2.90	15		1.99	13	
rgn	0.23	29		0.23	1	
roll3000	1 800.00	432 419	time	1 800.00	405 929	time
rout	35.23	24 218		34.93	23 948	
set1ch	0.49	9		0.54	13	

continued on next page

	*SCIP*_Default			SaP_Up		
Problem Name	T	nodes	Time limit	T	nodes	Time limit
seymour	1 800.00	34 678	time	1 800.01	32 036	time
seymour1	242.10	4 626		247.09	5 268	
sp97ar	1 800.00	12 358	time	1 800.01	11 783	time
stein27	0.83	3 963		0.80	3 963	
stein45	15.77	52 789		15.64	52 789	
stp3d	1 803.80	2	time	1 802.56	2	time
swath1	48.30	2 599		49.96	2 767	
swath2	51.28	2 802		44.21	2 107	
swath3	550.01	56 256		948.16	99 623	
t1717	1 800.20	625	time	1 800.01	644	time
timtab1	534.67	838 749		642.86	923 241	
timtab2	1 800.00	1 345 735	time	1 800.00	1 426 739	time
tr12-30	1 800.00	1 236 183	time	1 399.47	1 001 139	
vpm1	0.04	1		0.08	1	
vpm2	0.83	285		0.85	253	

Table 6.8: Results obtained with SCIP_Default and SaP_Up.

	SCIP_Default	SaP_Up			SaP_Rand		
Problem Name	t	t	SaP Obj	t_{SaP}	t	SaP Obj	t_{SaP}
10teams				0.03			0.04
30:70:4_5:0_5:100	0.04	0.04	1 339.0	0.66	0.03	1 569.0	0.71
30:70:4_5:0_95:98	0.04	0.05	1 331.0	0.68	0.04	1 579.0	0.72
30:70:4_5:0_95:100	0.05	0.04	1 267.0	0.68	0.05	1 459.0	0.74
a1c1s1	0.17	0.23		0.07	0.23		0.08
acc-0	0.72	0.81		0.01	0.95		0.02
acc-1	1.07	1.10		0.03	1.07		0.02
acc-2				0.02			0.03
acc-3				0.07			0.05
acc-4				0.05			0.07
acc-5				0.03			0.04
acc-6				0.05			0.05
aflow30a	4.69	0.09	4 606.0	0.02	5.34		0.02
aflow40b	15.47	0.86	8 300.0	0.09	15.55		0.13
air03	28.48	27.29		0.16	26.71	$8.3 \cdot 10^5$	0.06
air04				0.76			0.42
air05	6.52	8.48		1.91	6.64		0.19
arki001	2.42	2.48		0.07			0.03
atlanta-ip				3.31			2.68
bc1	4.94	5.93		1.01	6.09		1.11
bell3a	0.00	0.01	$1.6 \cdot 10^6$	0.01	0.01		0.00
bell5	0.01	0.01		0.00	0.01		0.01
bienst1	0.65	0.69		0.00	0.02	64.3	0.01
bienst2	0.86	1.12		0.00	0.02	72.3	0.00
binkar10_1				0.01			0.02
blend2	0.23	0.24		0.01	0.25		0.01
cap6000	0.26	0.35	−78 922.0	0.11	0.35	−90 093.0	0.12
dano3_3	33.25	23.29	583.5	22.91	8.46	583.0	8.10
dano3_4	29.84	6.07	586.9	5.62	7.11	583.3	6.66
dano3_5	33.54	5.37	588.5	4.85	6.36	584.4	5.86
dano3mip	189.62	2.23	801.9	0.36	2.21	879.5	0.38
danoint				0.01			0.01
dcmulti				0.01	0.03		0.00
disctom	2.77	12.24		9.37	7.52		4.69
ds		52.04	5 418.6	0.65	51.32	2 531.6	0.65
dsbmip	0.15	0.16		0.02	0.15		0.02

continued on next page

	SCIP_Default	SaP_Up			SaP_Rand		
Problem Name	t	t	SaP Obj	t_{SaP}	t	SaP Obj	t_{SaP}
egout	0.00	0.01	649.4	0.00	0.01	647.2	0.00
eilD76	1.77	1.75	3 630.9	0.01	1.74	2 210.8	0.00
enigma				0.01			0.00
fast0507	0.27	0.27	9 095.0	0.82	0.27	91 472.0	0.90
fiber	0.02	0.05		0.02	0.04		0.01
fixnet6	0.01	0.04		0.02	0.03	7 413.0	0.02
flugpl		0.00	$1.3 \cdot 10^6$	0.00			0.00
gen	0.02	0.02	$1.2 \cdot 10^5$	0.00	0.03	$1.2 \cdot 10^5$	0.00
gesa2-o	0.13	0.09	$1.1 \cdot 10^8$	0.03	0.19		0.06
gesa2	0.05	0.06	$1.3 \cdot 10^8$	0.03	0.06	$1.3 \cdot 10^8$	0.04
gesa3	0.05	0.05		0.04	0.06		0.03
gesa3_o	1.26	0.08	$1.4 \cdot 10^8$	0.03	1.29		0.03
glass4		0.69		0.00	0.63		0.00
gt2	0.01	0.01		0.00	0.00	$2.9 \cdot 10^5$	0.00
harp2	0.79	0.06	$-4.4 \cdot 10^7$	0.00	0.82		0.02
irp	7.25	3.61	18 142.1	0.08	3.55	14 632.0	0.09
khb05250	0.02	0.01	$1.6 \cdot 10^8$	0.00	0.02	$1.3 \cdot 10^8$	0.00
l152lav				0.03			0.01
liu	0.04	0.06	6 450.0	0.04	0.07	6 450.0	0.04
lrn		1.86	$1.9 \cdot 10^9$	0.27			0.22
lseu	0.01	0.01		0.00	0.01		0.00
manna81	0.01	0.01		0.10	0.02		0.09
markshare1	0.01	0.00	6 778.0	0.00	0.00	6 996.0	0.00
markshare1_1	0.00	0.01	5 877.0	0.00	0.01	5 877.0	0.00
markshare2	0.00	0.00	9 999.0	0.00	0.00	10 158.0	0.00
markshare2_1	0.00	0.00	8 450.0	0.00	0.01	8 450.0	0.00
markshare4_0	0.00	0.00	2 779.0	0.00	0.01	2 924.0	0.00
mas74	0.01	0.01	$1.6 \cdot 10^5$	0.00	0.02	$1.6 \cdot 10^5$	0.01
mas76	0.01	0.01	$1.6 \cdot 10^5$	0.00	0.01	$1.6 \cdot 10^5$	0.00
mas284	0.06	0.07	$1.6 \cdot 10^5$	0.01	0.07	$1.6 \cdot 10^5$	0.01
mik.250-20-75.1	0.00	0.00		0.02	0.01		0.03
mik.250-20-75.2	0.00	0.00		0.05	0.00		0.06
mik.250-20-75.3	0.00	0.01		0.02	0.00		0.04
mik.250-20-75.4	0.00	0.00		0.04	0.00		0.04
mik.250-20-75.5	0.00	0.00		0.05	0.00		0.07
misc03				0.01			0.00
misc06	0.05	0.04	13 951.9	0.01	0.05	13 951.9	0.01
misc07				0.03			0.00
mitre	5.64	5.67	$1.2 \cdot 10^5$	0.03	5.60	$1.4 \cdot 10^5$	0.04
mkc	0.01	0.02		0.02	0.01		0.03
mkc1	0.01	0.01	-277.8	0.04	0.01	-395.5	0.04
mod008	0.00	0.00	536.0	0.00	0.00	557.0	0.00
mod010	0.75	0.82		0.07	0.79		0.04
mod011	0.03	0.03		0.06	0.03		0.07
modglob	0.01	0.01	$3.6 \cdot 10^7$	0.00	0.01	$3.6 \cdot 10^7$	0.00
momentum1				2.42			1.21
momentum2				3.50			6.12
momentum3		585.54	$6.0 \cdot 10^5$	67.02	533.59	$5.0 \cdot 10^5$	16.63
msc98-ip				1.50			1.42
mzzv11	0.05	0.05	-100.0	0.25	0.05	-100.0	0.30
mzzv42z	0.06	0.06	-100.0	0.36	0.05	-10.0	0.47
neos1	0.09	0.10	140.0	0.03	0.09	131.0	0.02
neos2				0.10			0.05
neos3				0.17			1.03
neos4	2.50	2.48	$-4.8 \cdot 10^{10}$	0.01	2.52	$-4.8 \cdot 10^{10}$	0.02
neos5	0.00	0.00	23.0	0.00	0.00	18.0	0.00

continued on next page

	SCIP_Default	SaP_Up			SaP_Rand		
Problem Name	t	t	SaP Obj	t_{SaP}	t	SaP Obj	t_{SaP}
neos6				2.59			1.69
neos7	0.21	0.26		0.04	0.16	$2.6 \cdot 10^{7}$	0.05
neos8	0.15	0.16	−991.0	0.01	0.15	−400.0	0.01
neos9				1.54			1.62
neos10	16.14	16.26		0.03	16.15	−144.0	0.01
neos11				0.46			0.16
neos12				4.27			2.21
neos13	2.83	2.20	−28.0	0.67	2.17	−32.8	0.63
neos16				0.03			0.04
neos20				0.02			0.03
neos21	0.04	0.04	94.0	0.01	0.04	90.0	0.01
neos22	0.18	0.21	$1.3 \cdot 10^{6}$	0.06	0.21	$1.4 \cdot 10^{6}$	0.07
neos23	1.03	0.09	414.0	0.01	0.09	414.0	0.02
neos616206				0.02			0.02
neos632659	0.01	0.02		0.01	0.01	−74.0	0.01
neos648910	0.06	0.10	64.0	0.04	0.10		0.02
neos808444				0.89			1.13
neos818918		0.05	3 006.0	0.01	0.05	3 006.0	0.02
neos823206				0.20			0.18
neos897005		33.88		0.90	33.45		0.62
net12				0.23			0.28
noswot	0.01	0.01		0.01	0.01	−6.0	0.00
ns1648184		0.22	−1 216.1	0.01			0.01
ns1688347				0.06			0.02
ns1671066	0.19	0.21	232.5	0.05	0.21	220.0	0.06
ns1692855				0.13			0.05
nsrand-ipx	1.06	1.08	$1.7 \cdot 10^{5}$	0.13	1.04	$6.4 \cdot 10^{5}$	0.10
nug08	3.50	1.77	278.0	0.01	3.57		0.07
nw04	52.46	232.89	$1.4 \cdot 10^{5}$	182.97	51.62	62 704.0	2.86
opt1217	0.02	0.01	0.0	0.00	0.01	0.0	0.01
p0033	0.00	0.00		0.00	0.00		0.00
p0201		0.03	11 610.0	0.01			0.01
p0282	0.00	0.00	$8.0 \cdot 10^{5}$	0.00	0.00	$7.4 \cdot 10^{5}$	0.00
p0548	0.06	0.08		0.02	0.07		0.01
p2756	0.18	0.21	19 432.0	0.05	0.19	25 645.0	0.03
pk1	0.00	0.00	1 072.0	0.00	0.00	1 072.0	0.00
pp08a	0.00	0.01		0.00	0.01		0.00
pp08aCUTS	0.00	0.01		0.00	0.02		0.00
prod1	0.04	0.04	0.0	0.00	0.04	0.0	0.00
prod2	0.12	0.11	0.0	0.00	0.13	−1.0	0.01
protfold				0.06			0.04
qap10	28.51	8.74	468.0	0.01	28.68		0.18
qiu	0.06	0.08		0.01	0.07		0.00
qnet1	0.14	0.12	$4.3 \cdot 10^{5}$	0.01	0.13	$3.4 \cdot 10^{5}$	0.01
qnet1_o	0.06	0.05	$3.4 \cdot 10^{5}$	0.01	0.04	$3.6 \cdot 10^{5}$	0.01
ran8x32	0.01	0.03		0.02	0.01	8 639.0	0.01
ran10x26	0.01	0.03		0.02	0.02	7 267.0	0.01
ran12x21	0.01	0.02	6 613.0	0.01	0.03		0.01
ran13x13	0.00	0.01	5 147.0	0.01	0.01	6 063.0	0.01
rd-rplusc-21				0.51			0.27
rentacar				0.02			0.02
rgn	0.16	0.01	445.0	0.00	0.01	445.0	0.00
roll3000	3.77	3.81		0.06	3.87		0.11
rout	0.07	0.07		0.03	0.07		0.03
set1ch	0.02	0.03	$1.0 \cdot 10^{5}$	0.01	0.04	$1.2 \cdot 10^{5}$	0.02
seymour	0.01	0.01	1 278.0	0.02	0.00	1 272.0	0.02

continued on next page

Problem Name	SCIP_Default t	SaP_Up t	SaP Obj	t_{SaP}	SaP_Rand t	SaP Obj	t_{SaP}
seymour1	0.01	0.01	598.4	0.37	0.01	561.9	0.31
sp97ar	2.48	2.24	$7.8 \cdot 10^9$	0.28	2.19	$2.1 \cdot 10^{10}$	0.24
stein27	0.00	0.00	23.0	0.00	0.00	23.0	0.00
stein45	0.00	0.00	36.0	0.01	0.00	40.0	0.00
stp3d				318.72			164.82
swath		6.38	1 038.8	0.06	6.27	1 312.7	0.03
swath1				0.15			0.19
swath2				0.16			0.26
swath3				0.17			0.18
t1717				12.37			9.00
timtab1				0.00			0.01
timtab2				0.03			0.02
tr12-30	0.37	0.99		0.08	0.99		0.07
vpm1	0.01	0.01	25.0	0.00	0.01	24.0	0.00
vpm2	0.01	0.02		0.00	0.02		0.01

Table 6.9: results of the root node experiment for settings SCIP_Default, SaP_Rand and SaP_Up.

Problem Name	SaP_Down t	SaP Obj	t_{SaP}	SaP_None t	SaP Obj	t_{SaP}
10teams			0.03			0.04
30:70:4_5:0_5:100	0.04	729.0	0.65	0.05	1 987.0	0.78
30:70:4_5:0_95:98	0.03	1 008.0	0.65	0.05	2 172.0	0.79
30:70:4_5:0_95:100	0.04	631.0	0.68	0.03	2 043.0	0.83
a1c1s1	0.22		0.07	0.26		0.10
acc-0			0.10	0.05	0.0	0.01
acc-1	1.10		0.04	1.10		0.03
acc-2			0.05			0.03
acc-3			0.06			0.03
acc-4			0.08			0.06
acc-5			0.06			0.05
acc-6			0.07			0.04
aflow30a	0.09	4 606.0	0.01	0.09	4 606.0	0.01
aflow40b	0.86	8 300.0	0.08	0.86	8 300.0	0.08
air03	26.71	$6.4 \cdot 10^5$	0.06	26.59	$5.6 \cdot 10^5$	0.05
air04			0.26			0.06
air05	6.62		0.19	6.06	29 888.0	0.02
arki001			0.05	2.43		0.04
atlanta-ip			2.53			2.05
bc1	5.93		1.00	5.99		1.00
bell3a	0.00	$1.1 \cdot 10^6$	0.00	0.01	$1.7 \cdot 10^6$	0.00
bell5	0.01		0.00	0.01		0.00
bienst1	0.69		0.00	0.69		0.00
bienst2	1.12		0.01	1.12		0.01
binkar10_1	0.07	11 289.5	0.02	0.07	11 244.2	0.02
blend2	0.24		0.01	0.24		0.01
cap6000	0.36	−96 703.0	0.10	0.33	−99 268.0	0.11
dano3_3	7.71	583.8	7.32	23.84		0.11
dano3_4	5.55	586.0	5.11	44.61		0.13
dano3_5	11.96	588.1	11.45	37.86		0.14
dano3mip	2.19	807.6	0.36	188.16		0.25
danoint	0.03	66.5	0.01			0.01
dcmulti	0.04		0.00			0.01
disctom	12.24		9.37	12.28		9.39
ds	51.55	1 381.0	0.65	51.14	5 418.6	0.63

continued on next page

Problem Name	SaP_Down t	SaP Obj	t_{SaP}	SaP_None t	SaP Obj	t_{SaP}
dsbmip	0.15		0.02	0.16		0.03
egout	0.01	683.7	0.00	0.01	659.8	0.01
eilD76	1.75	2 019.7	0.01	1.74	3 630.9	0.01
enigma			0.00			0.00
fast0507	0.26	$1.2 \cdot 10^5$	0.84	0.26	$1.2 \cdot 10^5$	0.80
fiber	0.05		0.02	0.04		0.02
fixnet6	0.04		0.02	0.03		0.02
flugpl			0.00	0.00	$1.3 \cdot 10^6$	0.00
gen	0.02	$1.1 \cdot 10^5$	0.00	0.02	$1.1 \cdot 10^5$	0.00
gesa2-o	0.17		0.04	0.37		0.03
gesa2	0.05		0.03	0.06		0.03
gesa3	0.04		0.04	0.05		0.04
gesa3_o	0.97		0.03	1.29		0.02
glass4			0.00			0.01
gt2	0.02		0.00	0.02		0.00
harp2	0.83		0.03	0.80		0.01
irp	3.56	13 306.0	0.09	3.56	18 142.1	0.08
khb05250	0.02	$1.6 \cdot 10^8$	0.00	0.02	$1.6 \cdot 10^8$	0.00
l152lav			0.03			0.05
liu	0.07	6 450.0	0.05	0.08	6 450.0	0.04
lrn	33.02		0.21	1.96	$1.8 \cdot 10^9$	0.37
lseu	0.01		0.01	0.01		0.00
manna81	0.01		0.07	0.01		0.10
markshare1	0.00	7 142.0	0.00	0.00	7 022.0	0.00
markshare1_1	0.00	5 877.0	0.00	0.00	5 877.0	0.00
markshare2	0.00	10 335.0	0.00	0.00	10 233.0	0.00
markshare2_1	0.01	8 450.0	0.00	0.00	8 450.0	0.00
markshare4_0	0.01	2 961.0	0.00	0.00	2 852.0	0.00
mas74	0.02	$1.6 \cdot 10^5$	0.01	0.01	$1.6 \cdot 10^5$	0.00
mas76	0.02	$1.6 \cdot 10^5$	0.01	0.01	$1.6 \cdot 10^5$	0.00
mas284	0.06	$1.6 \cdot 10^5$	0.01	0.06	$1.6 \cdot 10^5$	0.01
mik.250-20-75.1	0.00		0.03	0.00		0.06
mik.250-20-75.2	0.00		0.02	0.00		0.02
mik.250-20-75.3	0.00		0.06	0.00		0.02
mik.250-20-75.4	0.00		0.01	0.00		0.04
mik.250-20-75.5	0.00		0.04	0.00		0.02
misc03			0.00			0.00
misc06	0.05	13 951.9	0.01	0.06	13 951.9	0.01
misc07			0.01			0.01
mitre	5.65	$1.3 \cdot 10^5$	0.04	5.73	$1.3 \cdot 10^5$	0.04
mkc	0.02	-1.8	0.02	0.01		0.30
mkc1	0.01	-401.2	0.03	0.02	-229.5	0.12
mod008	0.00	783.0	0.00	0.00	8 484.0	0.00
mod010	0.77		0.02	0.77		0.02
mod011	0.03		0.06	0.03		0.07
modglob	0.01	$3.6 \cdot 10^7$	0.01	0.01	$3.6 \cdot 10^7$	0.00
momentum1			0.75			1.06
momentum2			22.11			2.60
momentum3			71.95			94.23
msc98-ip			1.00			1.12
mzzv11	0.04	-100.0	0.34	0.05	-100.0	0.12
mzzv42z	0.06		0.46	0.06	-8.0	0.16
neos1	0.13		0.05	0.23		0.14
neos2			0.31			0.41
neos3			0.67			0.81
neos4	2.50	$-4.8 \cdot 10^{10}$	0.03	2.52	$-4.8 \cdot 10^{10}$	0.02

continued on next page

	SaP_Down			SaP_None		
Problem Name	t	SaP Obj	t_{SaP}	t	SaP Obj	t_{SaP}
neos5	0.00	23.0	0.00	0.00	23.0	0.00
neos6			11.66			2.10
neos7	0.17	$2.7 \cdot 10^7$	0.06	0.31		0.05
neos8	0.16	−1 284.0	0.01	0.14	−1 284.0	0.02
neos9			1.52			1.44
neos10	16.14	−194.0	0.01	16.00	−533.0	0.01
neos11			0.07			0.21
neos12			0.57			0.59
neos13	2.22	−28.0	0.67	2.22	−28.0	0.69
neos16			0.03			0.04
neos20			0.03			0.03
neos21	0.04	89.0	0.01	0.04	85.0	0.01
neos22	0.20	$1.3 \cdot 10^6$	0.06	0.21	$1.3 \cdot 10^6$	0.06
neos23	0.08	211.0	0.01	0.09	414.0	0.01
neos616206			0.02			0.03
neos632659	0.02		0.01	0.02		0.01
neos648910			0.01	0.08		0.01
neos808444			1.37			2.48
neos818918	0.06	3 006.0	0.02	0.05	3 006.0	0.01
neos823206			0.07			0.15
neos897005	48.99		16.62	35.11		2.31
net12			0.22			0.27
noswot	0.01	−7.0	0.01	0.01		0.00
ns1648184			0.01	0.21	−1 216.1	0.01
ns1688347			0.08			0.07
ns1671066	0.21	214.3	0.05	0.21	271.1	0.05
ns1692855			0.07			0.05
nsrand-ipx	1.05	$3.0 \cdot 10^6$	0.08	1.08	$1.4 \cdot 10^5$	0.11
nug08	1.78	272.0	0.00	1.78	272.0	0.01
nw04	48.98	19 470.0	0.57	50.49	36 512.0	1.01
opt1217	0.01	0.0	0.00	0.01	0.0	0.00
p0033	0.00		0.00	0.01		0.00
p0201	0.03	12 825.0	0.00	0.03	12 485.0	0.01
p0282	0.00	$8.4 \cdot 10^5$	0.01	0.00	$8.4 \cdot 10^5$	0.00
p0548	0.07		0.01	0.07		0.01
p2756	0.18	$1.3 \cdot 10^5$	0.02	0.20	$1.4 \cdot 10^5$	0.04
pk1	0.01	1 072.0	0.01	0.00	1 072.0	0.00
pp08a	0.00		0.00	0.00		0.00
pp08aCUTS	0.01		0.00	0.01		0.01
prod1	0.04	0.0	0.00	0.04	0.0	0.00
prod2	0.12	−1.0	0.00	0.12	0.0	0.01
protfold			0.02			0.09
qap10	8.67	454.0	0.01	8.81	454.0	0.02
qiu	0.09		0.01	0.07		0.01
qnet1	0.12	$2.7 \cdot 10^5$	0.01	0.13	$3.7 \cdot 10^5$	0.01
qnet1_o	0.05	$2.5 \cdot 10^5$	0.01	0.05	$2.8 \cdot 10^5$	0.01
ran8x32	0.04		0.02	0.04		0.02
ran10x26	0.03		0.02	0.03		0.02
ran12x21	0.01	6 613.0	0.01	0.02	6 613.0	0.01
ran13x13	0.01	5 147.0	0.01	0.01	5 147.0	0.01
rd-rplusc-21			0.42			5.36
rentacar			0.02			0.02
rgn	0.00	445.0	0.00	0.01	445.0	0.00
roll3000	3.85		0.09	3.83		0.05
rout	0.07		0.02	0.08		0.02
set1ch	0.03	$1.0 \cdot 10^5$	0.01	0.04	$1.0 \cdot 10^5$	0.02

continued on next page

Problem Name	SaP_Down t	SaP Obj	t_{SaP}	SaP_None t	SaP Obj	t_{SaP}
seymour	0.01	1 258.0	0.02	0.01	1 299.0	0.02
seymour1	0.01	565.0	0.31	0.01	587.0	0.31
sp97ar	2.19	$4.6 \cdot 10^{10}$	0.23	2.21	$7.5 \cdot 10^{9}$	0.25
stein27	0.00	23.0	0.00	0.00	23.0	0.00
stein45	0.00	38.0	0.00	0.00	39.0	0.00
stp3d			376.09			378.81
swath			0.22			0.07
swath1			0.11			0.14
swath2			0.12			0.15
swath3			0.13			0.16
t1717			0.94	3.41	$2.0 \cdot 10^{5}$	0.47
timtab1			0.00			0.01
timtab2			0.02			0.03
tr12-30	0.99		0.08	1.02		0.08
vpm1	0.01	21.0	0.00	0.01	23.0	0.00
vpm2	0.02		0.01	0.01		0.01

Table 6.10: results of the root node experiment for SaP_Down and SaP_None.

Bibliography

[AB07] Tobias Achterberg and Timo Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77 – 86, 2007. Mixed Integer Programming, IMA Special Workshop on Mixed-Integer Programming.

[ABCC06] David L. Applegate, Robert E. Bixby, Vasek Chvátal, and William J. Cook. *The Travelling Salesman Problem, A Computational Study*. Princeton University Press, 2006.

[Ach07] Tobias Achterberg. *Constraint Integer Programming*. PhD thesis, 2007.

[AKM06] Tobias Achterberg, Thorsten Koch, and Alexander Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):361–372, 2006.

[Apt03] Krzysztof R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.

[AS80] Bengt Aspvall and Richard E. Stone. Khachiyan's linear programming algorithm. *Journal of Algorithms*, 1(1):1 – 13, 1980.

[BCD+01] Egon Balas, Sebastián Ceria, Milind Dawande, Francois Margot, and Gábor Pataki. Octane: A new heuristic for pure 0-1 programs. *Operations Research*, 49(2):207–225, 2001.

[BCMS98] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. Savelsbergh. An updated mixed integer programming library: Miplib 3.0. *Optima*, 1998.

[Ber06] Timo Berthold. Primal heuristics for mixed integer programs. Master's thesis, 2006.

[BFL05] L. Bertacco, A. Fischetti, and A. Lodi. A feasible pump heuristic for general mixed integer programming. Technical report, DEIS–Università de Bologna, 2005.

[BGJ10] Ralf Borndörfer, Martin Grötschel, and Ulrich Jäger. *Production Factor Mathematics*. Springer, 2010.

[Chv83] Vasek Chvátal. *Linear Programming*. W. H. Freeman and Company, 1983.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction To Algorithms*. MIT Press, 2001.

[con] Concorde tsp solver. `http://www.tsp.gatech.edu/concorde.html`.

[cpl] ILOG CPLEX. `www-01.ibm.com/software/integration/optimization/cplex-optimizer`.

[DRL04] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102:71–90, 2004.

[ES00] H. A. Eiselt and C.-L. Sandblom. *Integer Programming and Network Models*. Springer, 2000.

[FGL05] Fishetti, Glover, and Lodi. The feasibility pump. *Mathematical Programming*, 2005.

[FL03] Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003. 10.1007/s10107-003-0395-5.

[KV02] Bernhard Korte and Jens Vygen. *Combinatorial Optimization Theory and Algorithms Second Edition*. Springer, 2 edition, 2002.

[Mar99] Alexander Martin. *Integer Programs with Block Structure*. PhD thesis, Technische Universität Berlin, 1999.

[Mit03] Hans Mittelmann. Decision tree for optimization software: Benchmarks for optimization software, 2003.

[Sch86] Alexander Schrijver. *Theory of integer and linear programming*. John Wiley and Sons, 1986.

[Wal09] Chris Wallace. ZI round, a MIP rounding heuristic. *Journal of Heuristics*, 16:715–722, 2009.

[Wol06] Kati Wolter. Implementation of cutting plane separators for mixed integer programs. Master's thesis, Technische Universität Berlin, 2006.